

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Plánování lidských zdrojů

Staffing

Zadání diplomové práce

Student: **Bc. Veronika Gavenčiaková**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Plánování lidských zdrojů
Staffing**

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je navrhnout a vyvinout interní software pro plánování lidských zdrojů v IT projektech. Bude se jednat o webovou aplikaci s možností integrace s ERP systémem (SAP By Design).

1. Analyzovat dostupné datové struktury v ERP systému (zaměstnanci, projekty, přiřazení) a navrhnout odpovídající doménový model.
2. Implementovat import dat z ERP systému.
3. Implementovat webové rozhraní aplikace, které musí umožňovat:
 - a) přehled aktuálních projektů a k nim přiřazených zaměstnanců,
 - b) přehled budoucích projektů,
 - c) přiřazování zaměstnanců na projekty,
 - d) přehled zaměstnanců bez projektu,
 - e) přehled zaměstnanců, kteří budou uvolnění v požadovaném období.
4. What-if analýza.
5. Grafické znázornění:
 - a) dostupnosti/přiřazení zaměstnanců,
 - b) trvání projektů.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 1. dubna 2016

.....
Gan

„Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a
zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.“

V Ostrave 18.03.2016


Vendavo CZ s.r.o.
Novinářská 7
709 00 Ostrava
IČ: 27077934
e-mail: infoCZ@vendavo.com
www.vendavo.cz



Na tomto mieste by som rada poďakovala Ing. Svatoplukovi Štolfovi, Ph.D. za jeho pomoc a odborné rady pri vypracovaní mojej diplomovej práce. Ďalej by chcela poďakovať všetkým, ktorí mi s prácou pomohli, pretože bez nich by táto práca nevznikla.

Abstrakt

Témou tejto diplomovej práce je navrhnuť interní software pre plánovanie ľudských zdrojov. Cieľom bolo vytvoriť webovú aplikáciu, určenú pre staffing manažérov, ktorých cieľom je efektívny a ekonomický chod firmy a to dôslednou organizáciou projektov a zamestnancov. Aplikácia umožňuje priradovanie zamestnancov na projekty, jednoduchý prehľad priradení zamestnancov na projekty, trvanie projektov a ďalších informácií potrebných pre čo najefektívnejšie plánovanie. Zároveň aplikácia umožňuje import dát z ERP systému. Najprínosnejšou funkcionalitou tejto aplikácie je What-if analýza, ktorá za použitia vhodného optimalizačného algoritmu hľadá čo najoptimálnejšie rozloženie zamestnancov na projekty.

Kľúčová slova: Plánovanie ľudských zdrojov, What-if analýza, webová aplikácia, ERP systém, Spring Boot, Java, React JS

Abstract

The topic of this thesis is an internal system design for human resources management. The aim was to create a web application for staff managers, who, by thorough organization of projects and employees, strive for effective and economical company operation. This application allows assignment of the employees to the specific projects, and provides for its overview. For the most effective planning, it also shows the project timing and other necessary information. Application's additional feature is the data import from the ERP system. The most important feature is What-if analysis, which, by using an appropriate optimisation algorithm, searches for the optimal project assignment distribution.

Key Words: Staffing, What-if analysis, ERP system, Spring Boot, Java, React JS

Obsah

| | |
|--|-----------|
| Seznam použitých zkratk a symbolů | 10 |
| Zoznam obrázkov | 11 |
| Zoznam tabuliek | 12 |
| 1 Úvod | 14 |
| 2 Ciele | 15 |
| 3 Prípadová štúdia | 16 |
| 4 Motivácia | 17 |
| 4.1 Pôvodné riešenie a jeho nedostatky | 17 |
| 4.2 K čomu bude nová aplikácia slúžiť | 18 |
| 5 Prehľad zamestnancov | 19 |
| 6 Projektové plánovanie | 20 |
| 7 What-if analýza | 25 |
| 8 Voľba algoritmu | 26 |
| 9 Genetický algoritmus | 27 |
| 9.1 Základné pojmy | 27 |
| 9.2 Princíp | 28 |
| 9.3 Operácie genetického algoritmu | 29 |
| 10 Využitie genetického algoritmu u What-if analýzy | 32 |
| 10.1 Výpočet fitness funkcie | 38 |
| 10.2 Generovanie alokácií | 41 |
| 10.3 Časová náročnosť genetického algoritmu | 42 |
| 11 Návrh a ďalšia implementácia | 44 |
| 11.1 ER diagram | 44 |
| 11.2 Technológie | 45 |
| 11.3 Integrácia s ERP systémom | 48 |
| 11.4 Kalendárový komponent | 50 |
| 11.5 Priradovanie zamestnancov na projekty | 52 |

| | |
|-------------------------|-----------|
| 12 Záver | 55 |
| Literatura | 57 |
| Přílohy | 57 |
| A Príloha na DVD | 58 |

Seznam použitých zkratk a symbolů

| | |
|---------|---|
| ERP | – Enterprise Resource Planning |
| JS | – JavaScript |
| XML | – Extensible Markup Language |
| AJAX | – Asynchronous JavaScript and XML |
| CSS | – Cascading Style Sheets |
| HTML | – HyperText Markup Language |
| HTTP | – Hypertext Transfer Protocol |
| DOM | – Document Object Model |
| JSON | – JavaScript Object Notation |
| REST | – Representational state transfer |
| API | – Application Programming Interface |
| HR | – Human Resources |
| ER | – Entity Relationship |
| CRUD | – Create, Read, Update, Delete |
| HATEOAS | – Hypermedia as the Engine of Application State |
| URI | – Uniform Resource Identifier |
| RSS | – Rich Site Summary |
| JSX | – Javascript XML |
| CSV | – Comma-separated values |

Zoznam obrázkov

| | | |
|----|--|----|
| 1 | Prehľad projektov | 17 |
| 2 | Alokácia zamestnanca Employee 1 | 19 |
| 3 | Možnosti filtrovania | 19 |
| 4 | Vytvorenie/modifikácia projektu | 21 |
| 5 | Chybová správa | 22 |
| 6 | Vytvorenie/modifikácia projektu | 23 |
| 7 | Prehľad priradení na projekt | 23 |
| 8 | Prehľad priradení na projekt | 24 |
| 9 | Vývojový diagram základného genetického algoritmu | 28 |
| 10 | Jednobodové kríženie | 30 |
| 11 | Viacbodové kríženie | 30 |
| 12 | Uniformné kríženie | 31 |
| 13 | Diagram aktivít genetického algoritmu | 32 |
| 14 | Pracovné pozície, ktoré potrebujeme obsadiť | 33 |
| 15 | Jedinec 1 | 34 |
| 16 | Jedinec 2 | 34 |
| 17 | Jedinec 3 | 35 |
| 18 | Jedinec 4 | 36 |
| 19 | Jedinec 5 | 37 |
| 20 | Jedinec 6 | 37 |
| 21 | Suboptimálne riešenie | 38 |
| 22 | Požiadavky | 41 |
| 23 | Výsledok optimalizácie | 41 |
| 24 | Nastavenie projektu | 42 |
| 25 | Výsledok optimalizácie | 42 |
| 26 | Časová náročnosť obsadenia 40tich pracovných pozícií | 43 |
| 27 | ER Diagram | 44 |
| 28 | Zamestnanec Employee1 | 46 |
| 29 | JSON zamestnanca Employee 1 | 47 |
| 30 | CSV | 49 |
| 31 | Harmonogram | 52 |
| 32 | Vytvorenie alokácie | 53 |

Zoznam tabuliek

| | | |
|---|---|----|
| 1 | Fitness hodnota počiatočnej populácie | 35 |
| 2 | Fitness hodnota prvej populácie | 36 |
| 3 | Časová náročnosť nájdenia výsledku | 42 |
| 4 | Import dát z ERP systému | 50 |
| 5 | Priradenie zamestnanca na projekt | 54 |
| 6 | Obsah CD | 58 |

Seznam výpisů zdrojového kódu

| | | |
|---|--|----|
| 1 | AJAX - načítanie všetkých zamestnancov | 45 |
| 2 | TimeLine | 51 |

1 Úvod

Prirodzeným procesom v prosperujúcej firme je jej neustály rozvoj a postupný rast. S pribúdajúcim množstvom nových projektov vzrastajú nároky na množstvo zamestnancov.

Microsoft Excel je tabuľkový nástroj, ktorý je finančne nenáročný, dobre dostupný, rozšírený a pomerne intuitívny, takže s ním vie pracovať veľa ľudí. Kvôli týmto výhodám býva často zvolený ako nástroj pre plánovania ľudských zdrojov. Na druhej strane Excel ako taký nie je nástroj priamo určený k plánovaniu. Nemá potrebné funkcie a preto je potrebné k nemu pristupovať veľmi zodpovedne.

Obzvlášť u väčšieho počtu zamestnancov a projektov už nastáva problém so zabezpečením konzistencie dát, udržania prehľadnosti a jednoduchosti plánovania a zdieľania informácií medzi viacerými manažérmi. V tejto chvíli vyvstáva problém, pretože Excel neumožňuje jednoduché zdieľanie a údržbu.

Tieto nedostatky bývajú motiváciou pre návrh intuitívnej webovej aplikácie umožňujúcej komfortnejší proces plánovania.

V mnohých firmách, veľkých či malých, sú projekty alokované nesystematicky. Chýba potenciál riadiť projekty s maximálnym vyťažením ľudí. Ľudia sú na projekty priradzovaní podľa aktuálnej dostupnosti sú aktuálne voľní. To potom zapríčiňuje, že pre nové projekty nemáme ľudí a musia sa zavádzať nadčasy. Frustrácia z nadčasov môže viesť k odchodu zamestnancov a tým firma príde o zdroje. Ďalšou alternatívou je odmietnutie projektu, čo znamená prísť o finančné zdroje.

Alokácia zamestnancov na projekty môže byť u vyššieho počtu projektov a zamestnancov pre človeka náročným procesom a preto je potrebné riešiť otázku ako proces plánovania vylepšiť a zefektívniť.

2 Ciele

Cieľom mojej práce je návrh a vytvorenie aplikácie pre plánovanie ľudských zdrojov.

Aplikácia je určená pre interné využitie firmou Vendavo s.r.o.

Medzi základné ciele aplikácie patrí nahradenie tabuľkového nástroja Microsoft Excel, ktorý je momentálne využívaný pre proces plánovania.

Aplikácia by mala ďalej proces plánovania uľahčiť a zjednodušiť. Tohto by sa malo docieľiť implementáciou nasledujúcich kľúčových funkcií medzi ktoré patrí:

- prehľad projektov a zamestnancov,
- prehľad alokácií na projekty,
- prehľad aktuálnych a budúcich projektov,
- prehľad zamestnancoch a ich aktuálnych a budúcich alokáciách,
- prehľad zamestnancov a ich dostupnosti v určitom období,
- grafické znázornenie trvania projektov a alokácii,
- import dát z ERP systému.

Kľúčovou funkcionalitou by mala byť implementácia What-if analýzy, ktorá bude za použitia vhodného algoritmu simulovať priradzovanie zamestnancov na projekty a pomôže včas odhaliť prípadné problémy v plánovaní.

Výstupom analýzy nemusí byť však len simulácia priradení na projekty, ale bude zároveň podporou procesu plánovania. Výsledky analýzy bude možné zmeniť na skutočné alokácie.

3 Prípadová štúdia

Vendavo je vedúcim poskytovateľom riešenia pre management a optimalizáciu cien a marží v oblasti B2B pre veľké obchodné spoločnosti po celom svete, ktoré pôsobia napríklad v oblasti high-tech odvetvia, chemického priemyslu, priemyselnej výroby či lekárskej techniky.

Produkty Vendava poskytujú spoločnostiam celofiremnú kontrolu nad procesom tvorby cien a prinášajú významné výsledky.

SAP, ktorý je popredným svetovým poskytovateľom podnikového softvéru, majúci viac ako 33,200 zákazníkov vo viac ako 120 krajinách je významným strategickým partnerom Vendava a využíva jeho komplexného riešenia správy cien a ziskov. Je predávaný po celom svete pod názvom SAP Price and Margin Management.

Softwarové riešenie od Vendava dokáže zvýšiť zisk až o 30% tým, že systematicky vylepšuje priemerne ceny v celom procese cenotvorby.

4 Motivácia

V tejto kapitole sa budeme venovať motivácii k vytvoreniu novej aplikácie.

Predstavíme základné nedostatky stávajúceho riešenia, spôsob akým nová aplikácia tieto problémy rieši a ďalšie informácie o pridanej hodnote aplikácie.

4.1 Pôvodné riešenie a jeho nedostatky

Správa zamestnancov a projektov je momentálne vykonávaná za použitia tabuľkového editoru Microsoft Excel. Toto riešenie pozostáva z dvoch hlavných kariet. Prvá karta poskytuje prehľad zamestnancov a ich alokácií na projekty. Druhá potom prináša pohľad na alokácie zo strany projektov.

Na Obr.1 môžete vidieť ukážku prehľadu projektov a k nim priradených zamestnancov.

| | | | | | | | | | | | | |
|--------------------------|---|--------|------------|---------------------------|--------|--------|----------------------------------|--|--|-----------------------|-----------------|--|
| January 15, 2014 | Active Projects | | | | | | | | | | | |
| Project | Dates | Region | PM | PC | SO | SA | LSE | SE | QA | Training Engagement | Comments | |
| 3M Global | 7/14/14 ends | West | | | Manish | Howard | | | | | UPDATED - 11/15 | |
| Air Products AMS | 1/1 - 6/30/2014 | East | | | | | | Bronislav | | | | |
| Analog - PM/DM | end 9/30/2014 | East | Chris K | Jin | Bob | Vinay | Praveen | Peter Gadula +1 3/15 | Ales Kosina | Christine (start 3/1) | UPDATED 1/17/14 | |
| Bayer | Thru 12/31/13 - Ends around Q3 2014 | EMEA | Gary (50%) | Johannes | Hans | Ales | | David Sedlik Marek Duciuc (moving to TI) Veronika Gavenciakova | Radim Vrkoslav Pavel Backa Vojtech Barta | | UPDATED 11/14 | |
| Boeing - PA | 1/13/2014 - end April | West | Venkat | Adrienne Ranjani (shadow) | Manish | Ram | | Lukas Sykora | Martin Absolin | | UPDATED - 01/17 | |
| Chevron R1 Stabilization | 12/13 - 3/31/14 | West | Najmul | | | | Shubha Bala Ruchira Mohapatra | Milan Matys | | | | |
| Chevron Release 2 | End Date - June 2014 | West | Najmul | Dawn | Manish | Sagar | Suma | | | | UPDATED - 12/6 | |
| Dupont (Danisco) | March 31, 2014 - End Date - Will Extend | EMEA | | | | | | Marek Vojtek | | | UPDATED - 11/21 | |

Obr. 1: Prehľad projektov

Prvým nedostatkom riešenia je, že na týchto kartách nie sú definované vzorce propagujúce informácie z jednej karty na druhú. Pri nedôslednom upravovaní dát, kedy človek modifikuje dáta iba na jednej karte dochádza k nekonzistencii dát a môžu tak nastať problémy pri priradzovaní ľudí na projekty. Buď môžeme priradzovať ľudí, ktorí už sú alokovaní na 100% alebo môžeme naopak stratiť informáciu o tom, že máme človeka, ktorý je v požadovanom období bez práce.

Ďalším nedostatkom riešenia je nedostatočná podpora importu dát z ERP systému. CSV súbory je samozrejme možné spravovať v tabuľkovom editore, ale problém nastáva v situácii, kedy nám z ERP systému prídu modifikované dáta. Zamestnanec môže byť povýšený, môže sa zmeniť jeho pracovná doba, atď. S rastúcim množstvom zamestnancov sa stáva manuálna úprava stávajúcich dát na základe nových dát z ERP značne nepohodlná, zdĺhavá a opäť môže nastať problém s nekonzistenciou dát.

Procesu projektového plánovania sa väčšinou zúčastňuje niekoľko manažérov. Každý z nich má vlastný xls súbor, s ktorým pracuje. V prípade, že by v súbore urobil zmenu, je potrebné, aby informoval všetkých kolegov o zmene, prípadne aby kolegom rozposlal aktuálnu verziu súboru.

Tieto nedostatky aktuálneho riešenia boli motiváciou k tomu, aby sa začalo uvažovať o vhodnejšom riešení.

Bolo rozhodnuté, že najvhodnejším riešením bude jednoduchá webová aplikácia, ktorá bude riešiť vyššie uvedené problémy.

Zároveň boli vznesené požiadavky na nové funkcionality, ktoré proces plánovania ešte viac zefektívnia.

Prvou požiadavkou bolo vytvorenie/použitie vhodnej kalendárovej komponenty, ktorá umožní grafický prehľad projektov a zamestnancov v čase.

Druhou požiadavkou bola implementácia What-if analýzy, ktorá umožňuje simuláciu priradzovania zamestnancov na projekty a ktorá včas odhalí prípadné problémy s nedostatkom zamestnancov.

4.2 K čomu bude nová aplikácia slúžiť

Primárnym cieľom aplikácie je zjednodušenie a zefektívnenie procesu plánovania ľudských zdrojov.

Po podpísaní nového projektu, prípadne po jeho predĺžení je potrebné zabezpečiť dostatok zamestnancov, ktorých môžeme na projekt priradiť. Dostatočný počet vhodne alokovaných zamestnancov vedie k úspešnému dokončeniu projektov v riadnych termínoch.

Zároveň je potrebné organizovať zamestnancov podľa skúseností či už s jednotlivými modulmi, tak aby nenastala situácia, že na projekte budú iba nováčikovia, alebo ľudia, ktorí s danou časťou aplikácie nemajú žiadne skúsenosti. Nedostatok alebo absencia skúsených zamestnancov môže viesť k nedodržaniu termínov a nevhodnému spracovaniu projektu. Toto môže viesť napríklad k výkonnostným problémom riešenia, prípadne k problémom s budúcimi zmenami projektu v rozširovaní stávajúcej funkcionality, či modulov.

Aplikácia bude taktiež umožňovať prehľad budúcich projektov. Staffing manažéri tak získajú dostatok času na vykonanie potrebných krokov k zabezpečeniu potrebného množstva zamestnancov. Môžu napríklad kontaktovať zamestnancov HR oddelenia s tým, že je potrebné prijať nových zamestnancov a zabezpečiť ich zaškolenie. Podporou pre nich bude What-if analýza, ktorá bude simulovať potenciálne rozmiestnenie zamestnancov na projekty a zároveň poskytne informáciu o hroziacom nedostatku zamestnancov.

Spolu s prehľadom budúcich projektov je dôležitý i prehľad aktuálnych projektov. Dostávame možnosť získať dostatočný prehľad o aktuálnom vyťažení firmy, či už z pohľadu zamestnancov tak z pohľadu projektov. Môžeme tvoriť lepšie stratégie a odhady pri plánovaní a dohadovaní ďalších projektov.

5 Prehľad zamestnancov

Z dôvodu lepšieho prehľadu o zamestnancoch je v aplikácii vytvorená obrazovka so základnými informáciami o zamestnancoch. Tu si môžeme vybrať konkrétneho zamestnanca, aby sme získali lepší prehľad o jeho situácii. Veľmi jednoducho tak zistíme, že daný človek je už dlho bez práce a že by bolo potrebné pokúsiť sa priradiť ho na nejaký projekt, či nájsť mu nejaký interný projekt v rámci firmy. Na Obr. 2 môžeme vidieť, že posledná alokácia zamestnanca Employee 1 na projekte skončila 31.08.2016 a odvtedy nepracoval na žiadnom inom projekte

| Project Name | Job | Skill | Knowledge % | Employee | Resource Location | Supply/Demand | Assignment % | Start Date | End Date |
|--------------|------------------------|--------------|--------------|------------|----------------------|----------------------|----------------------|------------|-----------|
| Please enter | Please enter a value | Please enter | Please enter | Employee | Please enter a value | Please enter a value | Please enter a value | | |
| Project 1 | Configuration Engineer | Deal manager | 100 % | Employee 1 | Offshore | Supply | 100 % | 4/1/2015 | 5/31/2015 |
| Project2 | Configuration Engineer | Deal manager | 100 % | Employee 1 | Offshore | Supply | 100 % | 6/1/2015 | 8/31/2015 |

Obr. 2: Alokácia zamestnanca Employee 1

Rovnako aplikácia umožňuje zobraziť si prehľad:

1. všetkých zamestnancov,
2. všetkých zamestnancov bez projektu v danom období,
3. všetkých zamestnancov s projektom v danom období.

The screenshot shows a web application interface for managing employees. At the top, there are buttons for 'Add Employee' and 'Import Employees'. Below these, there is a search bar and a filter section. The filter section includes a dropdown menu currently set to 'All employees', a date range selector with '4/26/2016' and 'To', and a green 'Filter' button. The dropdown menu is open, showing three options: 'All employees', 'Employees with projects', and 'Employees without projects'. Below the filter section, there is a table with columns: ID, Last Name, First Name, Rep, Country, Assignment %, Skills, and Projects. The first row of the table shows an employee with ID 12345, Last Name Alojs, First Name Jirasek, Rep GSD CZ Program Management, Country Germany, Assignment % 100, Skills Deal manager 100%, and Projects.

Obr. 3: Možnosti filtrovania

Za použitia tohto filtrovania môžeme získať napríklad prehľad o zamestnancoch bez projektu alebo o zamestnancoch, ktorí budú dostupní v požadovanom období. Táto informácia je užitočná pri procese plánovania budúcich projektov a pomôže nám v rozhodovaní o ďalších krokoch – akými sú napríklad podpísanie alebo nepodpísanie zmluvy alebo naberanie nových zamestnancov. Obdobné možnosti filtrovania sú dostupné aj v prehľade projektov:

1. všetkých projektov,
2. všetkých projektov bez zamestnancov v danom období,
3. všetkých projektov so zamestnancami v danom období.

Vďaka týmto možnostiam filtrovania môžeme získať napr. rýchly prehľad o aktuálnych projektoch a ich stave alebo o projektoch budúcich.

6 Projektové plánovanie

Pri priradovaní ľudí na projekty je potrebné brať do úvahy mnohé faktory, ktoré ovplyvňujú správne alokovanie ľudí na projekty. Preto je veľmi dôležité stanoviť si pri priradovaní ľudí na projekty všetky kritéria správneho priradenia na projekt.

Zamestnanec verzus projekt

Prvou vecou, ktorú si musíme stanoviť je, či je pre nás dôležité nájsť prácu čo najväčšiemu množstvu zamestnancov alebo či je pre nás prioritou obsadiť čo najviac projektov, či zaplniť jeden projekt na 100%. Na jednu stranu je dôležité, aby sme nemali človeka, ktorý je dlho bez práce, pretože sa môže začať nudiť, stratiť motiváciu a pre takéhoto zamestnanca potom nie je problém firmu opustiť.

Na druhú stranu je potrebné zaoberať sa otázkou obsadenia projektov. Zákazníka nechceme stratiť len z dôvodu, že nemáme dostatok zamestnancov. Vzhľadom na to, že väčšinu zákazníkov neuspokojí dodanie nekompletného tímu, s prísľubom priradenia ďalších ľudí v budúcnosti, je prioritou managementu obsadiť čo najviac projektov na 100% i za cenu toho, že budeme mať ďalší projekt, na ktorý nebudeme môcť priradiť nikoho.

Z dôvodu hroziaceho nedostatku zamestnancov je potrebné určiť si kritéria, podľa ktorých budeme určovať priority projektov.

Status

Prvým kritériom projektu, ktoré nás v rámci optimalizácie zaujíma je status. Status môže nadobúdať dva stavy, aktívny a potenciálny.

Aktívny znamená, že bola podpísaná zmluva so zákazníkom a na projekte sa už pracuje alebo je jeho začiatok naplánovaný.

Potenciálny projekt je v jednaní, zmluva ešte podpísaná nebola. Tento typ projektov nás zaujíma hlavne z dôvodu plánovania do budúcnosti. V prípade, že bude projekt podpísaný a stane sa aktívnym musíme zabezpečiť dostatok zamestnancov, čo môže znamenať, že budeme musieť zadať požiadavku oddeleniu ľudských zdrojov, aby začali prijímacie riadenie.

Z pohľadu optimalizácie nás ale zaujímajú aktívne projekty. Nemá zmysel naplánovať ľudí na projekty, ktoré sú zatiaľ v jednaní, v prípade, že máme aktívny projekt, ktorý začína za mesiac a na ktorý nemáme dostatok zamestnancov.

Čas

Ďalším kritériom, ktoré musíme brať do úvahy je čas. Pri hľadaní optimálneho riešenia je potrebné dať prioritu projektom, ktoré začínajú skôr. V našom prípade to znamená, že budú uprednostnené projekty, ktoré začínajú v nasledujúcich 3 mesiacoch.

Je samozrejmosťou že na projekty nemôžeme priradiť ľubovoľných zamestnancov.

Znalosti

Zamestnanci musia byť priradení na projekt na základe ich pracovnej pozície. Mimo toto kritérium je dôležité vziať do úvahy, aké znalosti má zamestnanec.

Musíme zamedziť situáciám, pri ktorých by sme na projekt priradili iba samých nováčikov alebo ľudí, ktorí nemajú s implementovaným modulom žiadne skúsenosti. Mohlo by to ohroziť priebeh projektu, od nedodržovania termínov až po nespokojnosť zákazníka, ktorá by mohla vyústiť až k ukončeniu vzájomnej spolupráce. Ideálnou situáciou je preto akási rovnováha v znalostiach ľudí priradených na jeden projekt tak, aby sme na projekte mali vždy niekoho skúsenejšieho, kto môže kolegov zaučiť a odovzdať im know-how.

Na to, aby sme ideálnu situáciu zabezpečili je potrebné pri vytváraní požiadaviek na jednotlivé pracovné pozície na projekte zároveň určiť, aké znalosti má mať vybraný zamestnanec a zároveň ohodnotiť úroveň konkrétnej znalosti.

Požadované znalosti a ich úroveň zadávame, spolu s ostatnými požiadavkami na pracovnú pozíciu, vo fáze vytvárania alebo úpravy projektu, ako môžeme vidieť na Obr. 4

| Job* | Skill* | Knowledge %* | Count* | Assignment %* |
|----------------------------|--------------|--------------|--------|---------------|
| Configuration Engineer | Deal manager | 100 % | 2 | 100 % |
| Quality Assurance Engineer | Deal manager | 100 % | 1 | 100 % |
| | | 0 % | 0 | 0 % |

Obr. 4: Vytvorenie/modifikácia projektu

Pracovná doba

Ďalším faktorom ovplyvňujúcim plánovanie je pracovná doba. Každý zamestnanec môže mať rôznu pracovnú dobu, pracovať napríklad iba 4 dni v týždni. Tento faktor musíme zohľadniť, aby pri plánovaní nenastala situácia, kedy je človek alokovaný na väčšie množstvo hodín, ako je jeho pracovný úväzok.

Jeden zamestnanec môže ďalej pracovať na väčšom množstve projektov súčasne, s tým, že úväzky na týchto projektoch nepresiahnu v súčte jeho celkový pracovný úväzok. Prekročenie

požadovanej pracovnej doby je síce možné riešiť nadčasom, ale situácia by nemala nastať pokiaľ to nie je nevyhnutne nutné. Preto musíme či už v rámci manuálneho plánovania alebo v rámci plánovania za pomoci algoritmu analýzy túto situáciu ošetriť. V prípade manuálneho priradzovania zamestnancov na projekt, je užívateľ upozornený chybovou správou ako môžete vidieť na Obr. 5.

The screenshot shows a 'Project Assignment' window with a red error banner at the top stating: 'Employee cannot be allocated from more than 100% of his assignment'. Below the banner, there are four labeled fields: 'Project*' with a dropdown menu showing 'Project2 (06/01/2016 - 09/30/2016)'; 'Job*' with a dropdown menu showing 'Configuration Engineer'; 'Skill*' with a dropdown menu showing 'Deal manager' and a text input field containing '100' followed by a '%' symbol; and 'Employee' with a dropdown menu showing 'Gavenciakova Veronika' and a close button (X).

Obr. 5: Chybová správa

Pracovná pozícia

Ďalším kritériom, dôležitým pre optimálne plánovanie je pracovná pozícia. Obmedzenie hovorí, že požiadavka na pracovnú pozíciu nebude delená medzi väčšie množstvo ľudí. Pokiaľ budeme potrebovať obsadiť pozíciu jedného vývojára na obdobie dvoch mesiacov musí túto pozíciu obsadiť jeden zamestnanec v plnom časovom rozsahu. Nie je možné pozíciu obsadiť dvomi zamestnancami, ktorí by si časový rozsah rozdelili.

Dôvodom je, že človek, ktorý už nejakú dobu na projekte pracuje, musí novému kolegovi odovzdať know-how. Proces odovzdávania informácii môže byť pre každý projekt iný, v závislosti na pracovnej pozícii, náročnosti projektu až po dobu, ktorú prvý kolega strávil na projekte. V praxi to znamená, že kolegovia by mali spoločne na projekte stráviť nejaký čas, počas ktorého prebehne odovzdanie znalostí. Vzhľadom na to, že nie sme schopný presne určiť čas, ktorý tento proces zaberie, bude z pohľadu programového riešenia jedna pracovná pozícia priradzovaná práve jednému zamestnancovi. Zároveň ale užívateľom aplikácie zostáva možnosť upraviť trvanie alokácie zamestnanca na projekte, prípadne vytvoriť ďalšiu pracovnú pozíciu a takýmto spôsobom napríklad rozdeliť jednu pracovnú pozíciu na dve.

Okrem pracovných pozícií, ktoré sú definované v rámci projektu môžeme definovať nejaké nad rámec projektu. Jedným z prípadov je vyššie spomínané rozdelenie jednej pozície na dve. Ďalším dôvodom môže byť, že v rámci definície projektu chceme uchovávať informáciu iba o skutočných požiadavkách zákazníka. Kým z pohľadu firmy môžeme chcieť napríklad pridať na projekt jedného človeka ako výpomoc. Častejšou situáciou je ale priradenie nováčika na projekt,

pretože práca na projekte je najlepším spôsobom toho, ako postupne spoznať celý systém a naučiť sa s ním pracovať.

Ako je vidieť na Obr. 6 pri vytváraní požiadaviek na projekt neurčujeme trvanie alokácií na jednotlivé pracovné pozície.

Project

Name* Project 1

Duration 4/1/2016 6/30/2016

Status* Active

Region* EMEA

| Job* | Skill* | Knowledge %* | Count* | Assignment %* |
|----------------------------|--------------|--------------|--------|---------------|
| Quality Assurance Engineer | Deal manager | 100 % | 1 | 100 % |
| Configuration Engineer | Deal manager | 100 % | 2 | 100 % |
| | | 0 % | 0 | 0 % |

Obr. 6: Vytvorenie/modifikácia projektu

Dátumy týchto alokácií môžeme opäť prepísať v sekcii priradzovania zamestnancov na projekty.

Na Obr. 7 môžeme vidieť príklad rozšíreného nastavenia projektu. Trvanie alokácie zamestnanca 1 bolo odvodené z projektu.

| | 2016 | | | | |
|------------|-------|----------------|----------------|------|------|
| | March | April | May | June | July |
| Employee 1 | | Project 1 100% | | | |
| Employee 2 | | Project 1 100% | | | |
| Employee 3 | | | Project 1 100% | | |

Obr. 7: Prehľad priradení na projekt

Zamestnanca 2 nepotrebujeme počas celého trvania projektu, jeho alokácia bola skrátená na mesiac. A Zamestnanec 3 je nováčik, ktorého potrebujeme zaškoliť, bol teda nad rámec projektu.

Jednotlivé zmeny v pracovných pozíciách môžete vidieť na Obr. 8

| Project Name ▾ | Job ▾ | Skill ▾ | Knowledge % ▾ | Employee ▾ | Resource Location ▾ | Supply/Demand ▾ | Assignment % ▾ | Start Date | End Date |
|----------------|------------------------|--------------|---------------|--------------|----------------------|----------------------|----------------------|------------|-----------|
| Please enter | Please enter a value | Please enter | Please enter | Please enter | Please enter a value | Please enter a value | Please enter a value | | |
| Project 1 | Configuration Engineer | Deal manager | 1 % | Employee 3 | Onshore | Supply | 100 % | 5/1/2016 | 6/30/2016 |
| Project 1 | Configuration Engineer | Deal manager | 100 % | Employee 2 | Offshore | Supply | 100 % | 4/1/2016 | 6/30/2016 |
| Project 1 | Configuration Engineer | Deal manager | 100 % | Employee 1 | Offshore | Supply | 100 % | 4/1/2016 | 6/30/2016 |

Obr. 8: Přehled přidání na projekt

7 What-if analýza

What-if (Čo-keď) analýza je analytická technika používaná pri rozhodovaní a riadení rizík. Základným princípom je hľadanie možných dopadov vybraných situácií.

Cieľom analýzy je overenie nečakaných potenciálnych rizík.

Jej výsledkom je zoznam nebezpečných situácií, ktoré môžu viesť k havarijným scenárom.

Včasným odhalením rizík je možné identifikovať nebezpečné situácie a vykonať všetky kroky, k tomu aby sme im predišli alebo aspoň eliminovali ich dopad.

V prípade plánovania ľudských zdrojov je cieľom analýzy vyhodnotiť potenciálne riziká spojené s uzavieraním zmlúv na nové projekty a fluktuáciou zamestnancov.

Pri nevhodnej organizácii zamestnancov sa môžeme dostať do situácie, kedy u podpísania nového kontraktu nebudeme mať zamestnancov, ktorých by sme na projekt mohli priradiť. Túto situáciu môžeme riešiť:

1. zavedením nadčasov,
2. prijatím nových zamestnancov,
3. neuzavretím zmluvy.

Všetky tieto riešenia ale majú nejaké nedostatky:

1. zamestnanci nemusia byť spokojní s tým že musia pracovať nadčasy a taktiež nemusia byť schopní to zvládať,
2. noví zamestnanci nemusia byť dostatočne vzdelaní v oblasti riešenej problematiky,
3. pri neuzatvorení zmluvy firma prichádza o zdroj príjmov, v prípade, kedy ide o stavajúceho zákazníka firma čelí riziku, že zákazníka definitívne stratí.

Výstupom analýzy by v našom prípade mal byť prehľad alokácií na projekty, ktorý nám poskytne informáciu o prípadných problémoch s alokovaním zamestnancov na projekty. Ďalej už je na firme ako sa k daným problémom postaví a ako prípadné nežiadúce situácie vyrieši.

Jej výstupom môže byť však aj bezproblémové priradenie zamestnancov na projekty.

Výstup analýzy nemusí mať len informatívny charakter. V prípade, kedy sme s jej výstupom spokojní môžeme si vybrať, ktoré alokácie chceme schváliť.

Aby boli zamestnanci na projekty rozdelení čo najoptimálnejšie je potrebné zvoliť vhodný optimalizačný algoritmus, ktorý určí čo najlepšie riešenie danej situácie.

8 Voľba algoritmu

Ako sme sa mohli presvedčiť v predchádzajúcej kapitole problém projektového plánovania a What-if analýzy, ktorá nám umožňuje akúsi simuláciu toho, čo nastane v budúcnosti, ak napríklad podpíšeme nový projekt, nemusí byť tak triviálny, ako sa na prvý pohľad môže zdať.

Aby sa tieto procesy zefektívnili a aby sme zamestnancov na projekty rozmiestňovali čo najsystematickejšie a najoptimálnejšie, bolo potrebné nájsť algoritmus, ktorý nám s procesom priradzovania zamestnancov na projekty pomôže.

V počiatočných fázach návrhu riešenia problému optimálneho riešenia problému What-if analýzy som uvažovala o použití nejakého knapsack algoritmu.

Keď si jednotlivé projekty predstavíme ako ruksaky, tak naším cieľom je naplniť ich zamestnancami tak, aby mali čo najmenšiu váhu ale zároveň najväčšiu hodnotu.

V našom prípade sú hodnotou zamestnanca jeho znalosti a jeho váhou jeho pracovná doba. Bohužiaľ u problému ruksaku som narazila na problém, že ruksaky nemusia vždy existovať v rovnakú dobu, nemusia byť vždy zákonite plnené z rovnakej skupiny zamestnancov.

Tým pádom nebolo možné použiť pre tento problém žiadnu ruksakovú metódu.

Ďalším kandidátom bola metóda hľadania s návratom, ktorá sa tiež nazýva metóda pokusov a opráv, metóda spätného prehľadávania či backtracking.

Ide o metódu systematického prehľadávania priestoru, v ktorom sa môže riešenie nachádzať. Jej hlavnou myšlienkou je postupná konštrukcia riešenia daného problému tak, aby boli ušetrené zbytočné kroky, ktoré nepovedú k riešeniu daného problému.

Nevýhodou riešenia je ale fakt, že pri vyššom počte požiadaviek na obsadenie pracovných pozícií a vysokom počte zamestnancov sa stáva táto metóda pomerne časovo náročnou.

Pri ďalšom skúmaní rôznych optimalizačných problémov sa ako ďalšia možnosť naskytla možnosť použiť genetické algoritmy, ktoré sú postavené na teórii prirodzenej selekcie a tom, že pokiaľ budeme v každej generácii kombinovať vždy iba najsilnejších jedincov, výsledný jedinci nasledujúcej generácie budú silnejší než ich rodičia.

Pri správnej implementácii toto riešenie dokáže vo vysokej miere eliminovať väčšinu nevhodných kandidátov riešenia a proces optimalizácie značne urýchliť.

Preto som sa problém optimálneho riešenia rozhodla vyriešiť práve za pomoci genetických algoritmov.

9 Genetický algoritmus

Pojem „genetický algoritmus“ bol zavedený v roku 1975 matematikom Johnom D. Hollandom. Ide o heuristický postup patriaci medzi evolučné algoritmy, ktoré čerpajú inšpiráciu v princípoch vývoja, ktoré sa uplatňujú v prírode.

Teoretickým podkladom pre genetické algoritmy je Darwinova teória prirodzenej selekcie. Podľa tejto teórie prežívajú silnejší jedinci, ktorí majú potom šancu preniesť svoju genetickú informáciu do ďalšej generácie. Selekcia najsilnejších jedincov je kľúčová pre optimalizáciu, pretože iteratívne dochádza k tomu, že máme množinu stále viac silnejších jedincov a tým aj potenciálnych kandidátov na hľadané riešenie.

Pomocou genetických algoritmov sa riešia zložité optimalizačné problémy, pre ktoré neexistuje použiteľný exaktný algoritmus.

Princípom genetického algoritmu je proces postupného vylepšovania populácie jedincov opakovaným aplikovaním genetických operátorov kríženia, mutácie a reprodukcie. Genetický algoritmus nemusí nutne nájsť najlepšie riešenie ale nájde také riešenie, ktoré nám postačuje, spĺňa nejaké podmienky alebo bolo nájdené v nami určenom konečnom čase. Tzn. blíži sa k nejakému extrému, ktorý je možné považovať za najlepšie riešenie. Pokiaľ nám dané riešenie nebude vyhovovať, môžeme genetický algoritmus nechať počítať ďalej.

Genetický algoritmus má ale i nevýhody.

Hlavnou nevýhodou je, že nemusí nutne nájsť najlepšie riešenie. Nájde však riešenie, ktoré sa k nemu približuje.

Ďalšou nevýhodou je nevhodnosť genetických algoritmov pre všeobecné riešenie problémov. Je nutné ich vždy prispôbiť pre konkrétnu doménu problému, tzn. vhodne navrhnúť hodnotiacu funkciu a genóm jedinca.

V prípade, že sa snažíme o návrh univerzálneho genetického algoritmu pre riešenie obecného problému, môže dôjsť k degenerácii a hľadaný výsledok nemusí byť nájdený v konečnom čase.

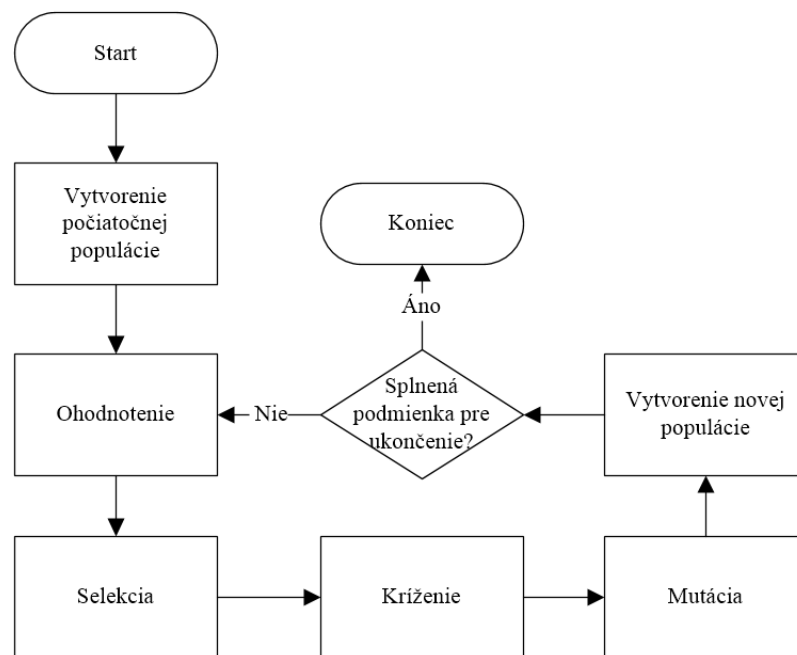
9.1 Základné pojmy

1. chromozóm je časť genetickej výbavy jedinca, ktorú je možné prenášať z rodiča na potomka. Najčastejšie je chromozóm reprezentovaný binárnym reťazcom, tzn. reťazcom núl a jednotiek,
2. gén je najmenšia časť chromozómu,
3. jedinec je nositeľom genetickej informácie. V praxi je pre nás jedinec potenciálnym kandidátom riešenia problému,
4. fitness hodnota určuje nakoľko je jedinec silný a tým pádom nakoľko je schopný prežiť,
5. populácia je skupina jedincov zastrešujúca jedincov spadajúcich do jednej generácie.

9.2 Princíp

Slovne môžeme algoritmus popísať takto:

1. Inicializácia – vytvorenie populácie z náhodne vygenerovaných jedincov.
2. Začiatok cyklu.
3. Ohodnotenie – určenie kvality/sily každého jedince v populácii, tzn. vyhodnotenie fitness funkcie $f(x)$.
4. Selekcia – výber jedincov s najväčšou hodnotou fitness funkcie.
5. Kríženie – kombináciou génov dvoch náhodne vybraných jedincov (rodičov) vznikne nový jedinec (potomok).
6. Mutácia – náhodná zmena časti jedince.
7. Koniec cyklu – ak nie je splnená podmienka pre ukončenie tak pokračuj od bodu 2.
8. Koniec algoritmu – výsledkom algoritmu je najsilnejší jedinec, reprezentujúci najlepšie riešenie.



Obr. 9: Vývojový diagram základného genetického algoritmu

9.3 Operácie genetického algoritmu

9.3.0.1 Inicializácia

Prvým krokom genetického algoritmu je vytvorenie počiatočnej populácie. Pri generovaní jedincov je možné využiť znalosť problému a vygenerovať takých jedincov, ktorí budú čo najbližšie optimálnemu riešeniu. Vo väčšine prípadov sa iníciaľna populácia vytvára náhodným výberom jedincov.

9.3.0.2 Selekcia

Úlohou selekcie je vybrať podskupinu jedincov populácie, ktorí budú ďalej predstavovať potenciálnych rodičov jedincov nasledujúcej generácie. Títo jedinci majú genetický predpoklad prežiť a predstavujú najvhodnejších kandidátov riešenia riešenej úlohy.

Väčšinou je cieľom selekcie vybrať najsilnejších jedincov a tým zabezpečiť neustály vývoj nasledujúcich populácií.

Metód výberu podskupiny jedincov je ale viac a nie každá zabezpečuje výber najsilnejších jedincov.

Typy selekcie

- Ruletová

Výber jedinca sa dá prirovnať k rulete, kde je každému jedincovi priradený výsek, ktorého veľkosť je úmerná fitness hodnote jedinca. Pri každom pokuse je potom vybraný jedinec, v ktorého výseku sa guľička zastaví.

- Turnajová selekcia

U turnajovej metódy dochádza k náhodnému výberu podskupiny jedincov. Podskupina musí mať aspoň dvoch jedincov. Víťazom turnaju je jedinec s vyššou fitness hodnotou.

- Náhodný výber

Ako už napovedá názov typu selekcie, ktorý vyberá jedincov bez ohľadu na fitness funkciu a tým pádom nezabezpečuje, že vyberieme podskupinu najsilnejších jedincov.

- Orezávanie

Je metóda, kedy jedincov zoradíme podľa fitness hodnoty od najsilnejšieho po najslabšieho a týchto jedincov rozdelíme na 2 časti. Tieto časti nemusia byť rovnako veľké, je to na nás ako populáciu rozdelíme. Ďalej pracujeme so skupinou silnejších jedincov, z ktorých páry rodičov vyberáme buď náhodne alebo nejakým deterministickým spôsobom.

9.3.0.3 Kríženie

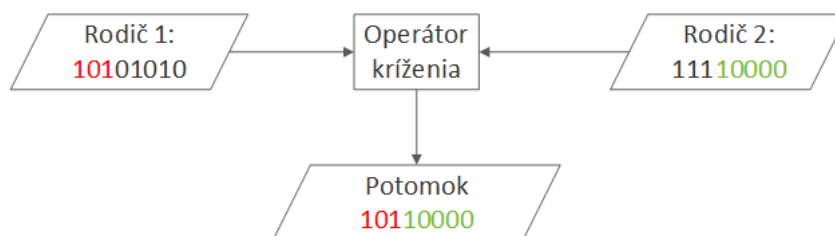
Kríženie je proces, kedy je vybraných niekoľko rodičov, väčšinou dvoch. Kombináciou ich génov vznikne nový jedinec. V ideálnom prípade by mal vzniknúť jedinec, ktorého fitness hodnota je vyššia než fitness hodnoty jeho rodičov.

Existuje niekoľko techník kríženia.

Základnými sú tieto 3:

- Jednobodové kríženie

Jednobodové kríženie je najjednoduchším spôsobom kríženia. V chromozóme dĺžky m vyberieme náhodne miesto n . Nový jedinec potom vznikne tak že sa spoja gény 1 až n z prvého jedinca a gény $n+1$ až m z druhého jedinca. Ukážku jednobodového kríženia môžeme vidieť na Obr. 10

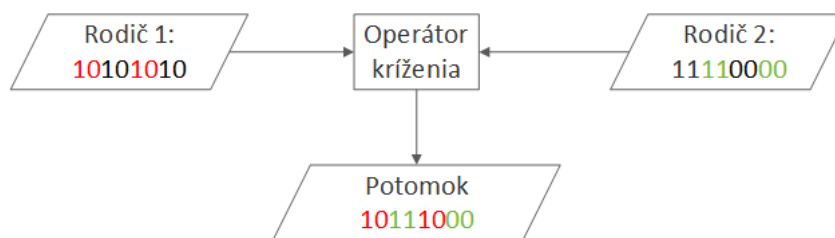


Obr. 10: Jednobodové kríženie

- Viacbodové kríženie

Princípom viacbodového kríženia je rozdelenie chromozómov na n častí. Chromozóm potomka sa následne skladá z nepárnych častí prvého rodiča a párnych častí druhého rodiča.

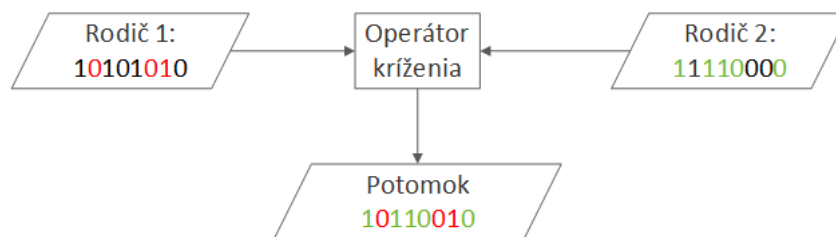
Ukážku viacbodového kríženia môžeme vidieť na Obr. 11



Obr. 11: Viacbodové kríženie

- Uniformné kríženie

Uniformné kríženie je založené na náhodnom výbere. U každého génu náhodne zvolíme, z ktorého rodiča sa daný gén vezme. Ukážku uniformného kríženia môžeme vidieť na Obr. 12



Obr. 12: Uniformné kríženie

9.3.0.4 Mutácia

Operácia mutácie je náhodná zmena náhodne vybraných jedincov.

Významným prínosom tejto operácie je fakt, že môže do danej generácie vnieť vlastnosť, ktorú žiaden jedinec tejto generácie nemá a tým pádom ju nemôže ďalej odovzdávať potomkom.

Mutácia sa kontinuálne snaží zachovať rôznorodosť jedincov v populácii. Klasická selekcia má za úlohu zachovávať populáciu najsilnejších jedincov a tým znižovať rôznorodosť jedincov v populácii.

Potlačenie rôznorodosti populácie môže spôsobiť predčasnú konvergenciu výpočtu k uviaznutiu k lokálnemu extrému. Preto sú mutácie nevyhnutnou súčasťou genetického algoritmu.

9.3.0.5 Výpočet fitness

Aby sme dokázali vybrať čo najoptimálnejšie riešenie je potrebné jedincov ohodnotiť, určiť takzvanú fitness hodnotu.

Hodnota fitness nám hovorí, ako sa dané riešenie približuje, respektíve vzdaluje od nami definovaného optimálneho riešenia.

Výpočet fitness hodnoty je úzko spätý s riešeným problémom. Napríklad v prípade riešenia problému priradenia zamestnancov na projekty, môže byť fitness hodnotou jedincov počet pracovných pozícií, ktoré sa nám podarilo obsadiť nejakým zamestnancom.

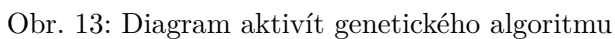
9.3.0.6 Ukončenie algoritmu

Genetický algoritmus končí vo chvíli, keď nájdeme jedinca, ktorý spĺňa naše očakávania.

Zároveň je nutné obmedziť dĺžku výpočtu algoritmu, aby sme predišli uviaznutiu v lokálnom extréme.

Pokiaľ nenájdeme optimálne riešenie algoritmus končí vo chvíli, keď dosiahneme maximálneho počtu generácií, ktorý sme si vopred stanovili.

Diagram aktivít znázorňujúci priebeh genetického algoritmu môžeme vidieť na Obr. 13



Aby problém nebol príliš komplikovaný, budem v príklade uvažovať iba jednu pracovnú pozíciu, Configuration Engineer. Všetky projekty budú mať požiadavky na ľudí s rovnakými znalosťami a rovnako všetci zamestnanci budú mať požadovanú znalosť.

Rozloženie pracovných pozícií, ktoré potrebujeme obsadiť, môžeme vidieť na Obr 14

Pracovné pozície, ktoré je potrebné obsadiť

| | 2016 | | | | | | | | | |
|---------------|------|------------------------|------------------------|-----|------------------------|------------------------|------------------------|---------------|-----|--|
| | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | |
| Project1 100% | | Configuration Engineer | | | | | | | | |
| Project1 100% | | Configuration Engineer | | | | | | | | |
| Project1 100% | | Configuration Engineer | | | | | | | | |
| Project2 50% | | | Configuration Engineer | | | | | | | |
| Project2 100% | | | Configuration Engineer | | | | | | | |
| Project3 50% | | | | | Configuration | | | | | |
| Project4 100% | | | | | | Configuration Engineer | | | | |
| Project5 100% | | | | | | | Configuration Engineer | | | |
| Project6 100% | | | | | | Configuration Engineer | | | | |
| Project7 100% | | | | | | | | Configuration | | |
| Project8 100% | | | | | Configuration Engineer | | | | | |

Obr. 14: Pracovné pozície, ktoré potrebujeme obsadiť

Počiatočná populácia

V prvom kroku si vytvoríme počiatočnú populáciu.

Ako bolo vysvetlené v predchádzajúcej kapitole, populácia je zložená z jedincov. V prípade uvažovaného problému priradovania zamestnancov na projekty, nie sú jedincami jednotlivé pracovné pozície, ale harmonogramy priradení zamestnancov na pracovné pozície.

Výstupom a hľadaným riešením je v našom prípade harmonogram. Príkladom takého priradenia je rozloženie pracovných pozícií na Obr. 4.

Pri vytváraní populácie sa snažíme obsadiť všetky pracovné miesta náhodne vybraným zamestnancom. Toto náhodne priradenie ale musí byť platné, to znamená že priradený zamestnanec:

- musí byť dostupný v požadovanom období a zároveň po priradení na projekt nemôže jeho pracovná doba prekročiť jeho pracovný pomer. To znamená že človek, ktorý je zamestnaný na 3 dni (60%) v týždni nemôže byť zamestnaný na 5 dní (100%) v týždni,

- musí byť zamestnaný na požadovanej pracovnej pozícii,
- musí mať požadovanú znalosť a musí ju ovládať minimálne na toľko percent, koľko je v požiadavkách na pracovnú pozíciu,
- zamestnanec môže v danom období pracovať na viac než jednom projekte. Pri vytváraní počiatočnej populácie, ale priradujeme daného zamestnanca práve jeden krát.

V ukážke vygenerujeme jednoduchú populáciu, skladajúcu sa iba z 3 jedincov, ktorých môžete vidieť na Obr. 15, Obr. 16 a Obr. 17

| | 2016 | | | | |
|---------------|------------|------------|-----|-----|-----|
| | May | Jun | Jul | Aug | Sep |
| Project1 100% | Employee 1 | | | | |
| Project1 100% | Employee 3 | | | | |
| Project1 100% | Employee 2 | | | | |
| Project2 50% | | Employee 4 | | | |
| Project2 100% | | Employee 5 | | | |

Obr. 15: Jedinec 1

| | 2016 | | | | | | |
|---------------|------------|-----|------------|------------|------------|------------|-----|
| | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| Project2 100% | Employee 5 | | | | | | |
| Project5 100% | | | | | Employee 4 | | |
| Project6 100% | | | | Employee 3 | | | |
| Project7 100% | | | | | | Employee 2 | |
| Project8 100% | | | Employee 1 | | | | |

Obr. 16: Jedinec 2

Ďalším krokom genetického algoritmu je ocenenie jednotlivých jedincov. Pre zjednodušenie bude v tomto príklade fitness hodnotou jedinca percentuálne vyjadrenie úspešne obsadených pracovných pozícií k celkovému počtu pozícií, ktoré potrebujeme obsadiť

n =počet úspešne obsadených pozícií

m =celkový počet pozícií, ktoré potrebujeme obsadiť

$$fitness(x) = \frac{n}{m} * 100 \quad (1)$$

| | 2016 | | | | | | |
|---------------|------------|-----|------------|------------|-----|------------|-----|
| | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| Project2 100% | Employee 1 | | | | | | |
| Project2 50% | Employee 3 | | | | | | |
| Project3 100% | | | Employee 4 | | | | |
| Project6 100% | | | | Employee 2 | | | |
| Project7 100% | | | | | | Employee 5 | |

Obr. 17: Jedinec 3

Pre výpočet fitness hodnoty je potrebné poznať počet pracovných pozícií, ktoré potrebujeme obsadiť. V našom prípade je počet týchto pozícií 11. Fitness hodnota jednotlivých jedincov môžeme vidieť v Tabuľka 1:

| Fitness | % |
|-----------|-----|
| Jedinec 1 | 45% |
| Jedinec 2 | 45% |
| Jedinec 3 | 45% |

Tabuľka 1: Fitness hodnota počiatočnej populácie

Po vygenerovaní skontrolujeme, či počiatočná populácia neobsahuje jedinca, ktorého fitness hodnota by bola 100%. Čiže overíme či sme už pri generovaní iniciálnej populácie nevytvorili jedinca, ktorý spĺňa všetky podmienky optimálneho priradenia zamestnancov na projekty.

Ak populácia neobsahuje dokonalého jedinca, vezmeme najlepšieho jedinca a uložíme si ho do pomocného objektu pre ďalšie použitie. V našom prípade majú všetci traja rovnakú fitness hodnotu. Uložíme si teda prvého z nich. Pokračujeme ďalším krokom, ktorým je selekcia, čiže výber najsilnejších jedincov. V mojom algoritme som si zvolila selekciu orezávaním, to znamená, že jedincov zoradím podľa ich fitness hodnoty a vyberiem podskupinu najlepších. V mojom prípade vyberáme 50% jedincov z predchádzajúcej populácie.

Tento krok v príklade vynecháme a naďalej budeme pracovať s tromi jedincami a prejdeme k vytváraniu novej generácie za použitia genetických operátorov kríženia a mutácie.

Proces kríženia a mutácie opakujeme dovtedy, kým nová populácia neobsahuje toľko jedincov ako predchádzajúca. V ukázkovom príklade ale budem pokračovať iba s dvoma najsilnejšími jedincami.

Náhodne vyberieme 2 jedincov, rodičov, zo stávajúcej populácie a budeme ich krížiť. Kríženie v našom prípade funguje na základe náhodného výberu pracovných pozícií a ich priradenie do nového jedinca, potomka. Pri priradovaní pozícií do potomka musíme opäť dbať na to, aby výsledný jedinec spĺňal všetky podmienky validného jedinca. Pozíciu, ktorú sa snažíme pridať do potomka vždy z rodiča odstránime, bez ohľadu na to, či sa nám pozíciu do nového jedinca

podarilo vložiť alebo nie. Proces vkladania opakujeme dovtedy, kým sú v rodičoch pracovné pozície, ktoré sme do potomka ešte neskúšali vložiť.

Po každom vytvorení nového jedinca nasleduje proces mutácie. Mutácia nie je častý jav, preto na základe náhodne vypočítaného čísla určujeme či nového jedinca zmutujeme alebo nie.

V mojej implementácii genetického algoritmu mutujeme každého jedinca, u ktorého náhodným výberom vyberiem z čísel od 0 do 1000 číslo menšie alebo rovné číslu 20.

Zmutovať jedinca môžeme dvoma spôsobmi:

- odstránenie náhodne vybranej pracovnej pozície,
- pridanie novej pracovnej pozície. Zamestnanca pre novú pracovnú pozíciu opäť vyberáme náhodne, zo všetkým dostupných zamestnancov splňujúcich kritéria.

Či bude pracovná pozícia pridaná alebo odobraná je opäť dané náhodným výberom. Na Obr. 9 môžete vidieť že u jedinca 5 došlo k obom typom mutácie. Odobranie zamestnanca Employee 5 z Projectu 7 a zároveň úspešné pridanie Employee 5 na Project 2.

1. generácia

Potomkom jedincov 1 a 2 je jedinec 4, ktorého môžete vidieť na Obr. 18 a výsledkom kríženia jedincov 2 a 3 a následnej mutácie je jedinec 5 (Obr. 19).

| | 2016 | | | | | | | |
|---------------|------------|------------|-----|-----|------------|------------|------------|-----|
| | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| Project1 100% | Employee 3 | | | | | | | |
| Project1 100% | Employee 2 | | | | | | | |
| Project1 100% | Employee 1 | | | | | | | |
| Project2 50% | | Employee 4 | | | | | | |
| Project2 100% | | Employee 5 | | | | | | |
| Project5 100% | | | | | | Employee 4 | | |
| Project6 100% | | | | | Employee 3 | | | |
| Project7 100% | | | | | | | Employee 2 | |

Obr. 18: Jedinec 4

Opäť ohodnotíme jedincov novej populácie. Vypočítané hodnoty môžeme vidieť v Tabuľka 2:

| Fitness | % |
|-----------|-----|
| Jedinec 4 | 82% |
| Jedinec 5 | 64% |

Tabuľka 2: Fitness hodnota prvej populácie

| | 2016 | | | | | |
|---------------|--|-----|------------|------------|--|--|
| | Jun | Jul | Aug | Sep | | |
| Project2 100% | Employee 5 | | | | Zamestnanec pridaný v rámci mutácie | |
| Project3 50% | | | Employee 4 | | | |
| Project4 100% | | | | Employee 2 | | |
| Project5 100% | | | | Employee 4 | | |
| Project6 100% | Zamestnanec odobraný v rámci mutácie | | | Employee 3 | | |
| Project7 100% | | | | | Employee 5 | |
| Project8 100% | | | Employee 1 | | | |

Obr. 19: Jedinec 5

Ako môžeme vidieť v tabuľke, ani jeden jedinec nie je optimálnym riešením. V tomto prípade vezmeme najlepší z nich a porovnáme ho s najlepším jedincem z predchádzajúcej generácie a ak je fitness hodnota vyššia nahradíme ním predchádzajúceho najsilnejšieho jedinca. V našom prípade má jedinec 4 vyššiu hodnotu ako predchádzajúci najlepší jedinec, takže si uložíme novú hodnotu 82% a pokračujeme vytváraním ďalšej generácie.

Vytvoríme opäť novú populáciu, skladajúcu sa z 3 potomkov. A overíme či pri krížení a mutácii náhodou nevznikol jedinec, ktorý by reprezentoval optimálne riešenie. Na Obr. 9 môžeme vidieť nového jedinca, ktorý vznikol skrížením jedincov 4 a 5. Tento jedinec má obsadené všetky pracovné pozície, tzn. Našli sme optimálne riešenie a nie je potrebné ďalej vytvárať nové generácie a týmto môžeme algoritmus ukončiť.

| | 2016 | | | | | | | |
|---------------|------------|------------|-----|------------|------------|------------|------------|-----|
| | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| Project1 100% | Employee 3 | | | | | | | |
| Project1 100% | Employee 2 | | | | | | | |
| Project1 100% | Employee 1 | | | | | | | |
| Project2 100% | | Employee 5 | | | | | | |
| Project2 50% | | Employee 4 | | | | | | |
| Project3 50% | | | | Employee 4 | | | | |
| Project4 100% | | | | | Employee 2 | | | |
| Project5 100% | | | | | | Employee 4 | | |
| Project6 100% | | | | | Employee 3 | | | |
| Project7 100% | | | | | | | Employee 5 | |
| Project8 100% | | | | Employee 1 | | | | |

Obr. 20: Jedinec 6

Suboptimálne riešenie

Vo vyššie popísanom príklade sme našli optimálne riešenie. Výsledkom genetického algoritmu, ale nemusí byť vždy výsledkom úspešné obsadenie všetkých pracovných pozícií. Príčinou je nedostatok zamestnancov spĺňajúcich požiadavky.

Aby nám proces vytvárania nových generácií, v snahe nájsť 100% obsadenie pracovných pozícií, neskončil v nekonečnom cykle, pridávame do ukončovacej podmienky genetického algoritmu maximálny počet generácií, ktoré majú byť vytvorené.

Ďalším krokom by mohlo byť opätovné spustenie genetického algoritmu so zníženými nárokmi na obsadenosť pracovných pozícií, takže by sme upustili od pôvodného požiadavku mať naplnené všetky a skúsili obsadiť napr. iba 90%.

Namiesto opakovaného spúšťania algoritmu s postupným znižovaním nárokov môžeme využiť možnosť udržiavať si informáciu o najlepšom jedincovi skrz všetky populácie.

Túto informáciu sme si počas prevádzania algoritmu ukladali bokom.

Výsledkom môže byť napríklad jedinec, ktorý je zobrazený na Obr. 21. Pracovné pozície, ktoré sa nám nepodarilo obsadiť sú označené červenou farbou.

| | 2016 | | | | | | | | | |
|---------------|------|------------------------|------------|-----|------------------------|------------|------------------------|------------------------|-----|--|
| | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | |
| Project1 100% | | Employee 2 | | | | | | | | |
| Project1 100% | | Employee 5 | | | | | | | | |
| Project1 100% | | Configuration Engineer | | | | | | | | |
| Project2 50% | | | Employee 4 | | | | | | | |
| Project2 100% | | | Employee 1 | | | | | | | |
| Project3 50% | | | | | Configuration Engineer | | | | | |
| Project4 100% | | | | | | Employee 1 | | | | |
| Project5 100% | | | | | | | Configuration Engineer | | | |
| Project6 100% | | | | | | Employee 6 | | | | |
| Project7 100% | | | | | | | | Configuration Engineer | | |
| Project8 100% | | | | | Employee 3 | | | | | |

Obr. 21: Suboptimálne riešenie

10.1 Výpočet fitness funkcie

Dôležitosť určenia správneho výpočtu fitness funkcie, je najlepšie vidieť na prípade, kedy neexistuje také riešenie ktoré by viedlo k 100% optimálnemu výsledku.

V našom prípade ide o situáciu, kedy nemáme dostatok zamestnancov na to, aby sme ich rozmiestnili na všetky požadované pracovné pozície. Z toho dôvodu je si potrebné určiť váhy jednotlivých riešení.

Prvým kritériom ovplyvňujúcim fitness funkciu jedinca je status projektu.

Ako už bolo spomenuté v kapitole projektové plánovanie, projekt sa môže nachádzať v jednom z dvoch stavov: aktívny alebo potenciálny.

Z hľadiska plánovania má vyššiu prioritu aktívny projekt.

Projekty s potenciálnym stavom sa začínajú zamestnancami zaplňať až v prípade, kedy pre zamestnancov neexistuje vhodná pracovná pozícia na niektorom z aktívnych projektov.

Toto kritérium je zároveň najdôležitejším vo výpočte celkovej hodnoty fitness jedinca.

Jedinci populácie s vyšším počtom aktívnych projektov musia byť uprednostnené pred jedincami s nižším množstvom aktívnych projektov.

Prvú zo zložiek, ktorá bude určovať výslednú hodnotu fitness jedinca teda určíme ako pomer aktívnych projektových pozícií voči celkovému počtu pracovných pozícií ktoré potrebujeme obsadiť.

Hodnotu zložky vypočítame podľa vzorca:

n =počet aktívnych pracovných pozícií

m =celkový počet pozícií, ktoré potrebujeme obsadiť

$$váha(x) = \frac{n}{m} * 100 \quad (2)$$

Ďalším kritériom je dátum začiatku projektu. Je dôležité uprednostniť projekty, ktoré začínajú skôr.

Jedinci s vyšším počtom pracovných pozícií začínajúcich v priebehu nasledujúcich 90 dní majú vyššiu prioritu než pracovné pozície s nižším počtom pracovných pozícií začínajúcich v tomto období.

Druhú zo zložiek teda získam podľa vzorca:

n =počet pracovných pozícií začínajúcich v priebehu nasledujúcich 90 dní

m =celkový počet pozícií, ktoré potrebujeme obsadiť

$$váha(x) = \frac{n}{m} * 100 \quad (3)$$

Posledným kritériom je pomer počtu pracovných pozícií, ktoré sa nám podarilo obsadiť voči celkovému počtu pracovných pozícií.

Túto zložku vypočítame podľa vzorca:

n =počet pracovných pozícií, ktoré sa nám podarilo obsadiť

m =celkový počet pozícií, ktoré potrebujeme obsadiť

$$váha(x) = \frac{n}{m} * 100 \quad (4)$$

Celkovú hodnotu fitness jedinca je potrebné určiť na základe troch získaných zložiek. Hodnota fitness musí byť vyjadrená jednou číselnou hodnotou. S číslom, vyjadrujúcim hodnotu fitness budeme teda pracovať ako s reťazcom deviatich znakov. Prvé tri znaky reťazca obsahujú hodnotu

zložky, ktorá zohľadňuje status projektu. Nasledujú tri znaky pre hodnotu zložky popisujúcej dátum začiatku projektu. Posledné tri znaky sú určené pre zložku určujúcu obsadenosť projektov.

10.1.0.1 Príklad 1:

celkový počet pozícií, ktoré potrebujeme obsadiť $k = 10$

počet aktívnych pracovných pozícií, $l = 4$

$$váha(l) = \frac{l}{k} * 100 = \frac{4}{10} * 100 = 40 \quad (5)$$

počet pracovných pozícií začínajúcich v priebehu nasledujúcich 90 dní, $m = 6$

$$váha(m) = \frac{m}{k} * 100 = \frac{6}{10} * 100 = 60 \quad (6)$$

počet pracovných pozícií, ktoré sa nám podarilo obsadiť, $n = 8$

$$váha(n) = \frac{n}{k} * 100 = \frac{8}{10} * 100 = 80 \quad (7)$$

Výsledný reťazec teda bude 04060080 a následne je výsledná hodnota fitness funkcie 40060080.

10.1.0.2 Príklad 2:

celkový počet pozícií, ktoré potrebujeme obsadiť $k = 10$

počet aktívnych pracovných pozícií, $l = 0$

$$váha(l) = \frac{l}{k} * 100 = \frac{0}{10} * 100 = 0 \quad (8)$$

počet pracovných pozícií začínajúcich v priebehu nasledujúcich 90 dní, $m = 0$

$$váha(m) = \frac{m}{k} * 100 = \frac{0}{10} * 100 = 0 \quad (9)$$



































počet pracovných pozícií, ktoré sa nám podarilo obsadiť, $n = 7$

$$váha(n) = \frac{n}{k} * 100 = \frac{7}{10} * 100 = 70 \quad (10)$$

Výsledný reťazec teda bude 00000070 a následne je výsledná hodnota fitness funkcie 70

10.1.0.3 Ukážka z aplikácie

Na Obr. 22 môžeme vidieť nadefinované projekty. Projekty 2 a 4 majú status potenciálny, všetky ostatné sú aktívne.

| | Name <small>▼</small> | Region <small>▼</small> | Status <small>▼</small> | Start Date | End Date | Jobs |
|---|-----------------------|-------------------------|-------------------------|------------|------------|--|
| | <small>Please</small> | <small>Please</small> | <small>Please</small> | | | |
|   | Project1 | EAST | Active | 5/1/2016 | 7/31/2016 |  2  100%: Configuration Engineer  Deal manager 100%  |
|   | Project2 | EMEA | Potential | 6/1/2016 | 9/30/2016 |  1  50%: Configuration Engineer  Deal manager 100%   1  100%: Configuration Engineer  Price manager 100%  |
|   | Project3 | EAST | Active | 8/1/2016 | 9/30/2016 |  1  50%: Configuration Engineer  Deal manager 100%  |
|   | Project4 | EMEA | Potential | 8/1/2016 | 12/31/2016 |  1  100%: Configuration Engineer  Deal manager 100%  |
|   | Project5 | EMEA | Active | 9/1/2016 | 12/31/2016 |  1  100%: Configuration Engineer  Deal manager 100%  |

Obr. 22: Požiadavky

Na Obr. 23 môžeme vidieť výsledok optimalizačného algoritmu, ktorý uprednostnil aktívne projekty a teda projekty 2 a 4 zostali bez zamestnancov.

| | 2016 | | | | | | | | | |
|---------------|-------|------------------------|------|------|------------------------|------------|---------|----------|----------|--|
| | April | May | June | July | August | September | October | November | December | |
| Project1 100% | | Employee 1 | | | | | | | | |
| Project1 100% | | Employee 2 | | | | | | | | |
| Project2 100% | | Configuration Engineer | | | | | | | | |
| Project2 50% | | Configuration Engineer | | | | | | | | |
| Project3 50% | | | | | Employee 1 | | | | | |
| Project4 100% | | | | | Configuration Engineer | | | | | |
| Project5 100% | | | | | | Employee 2 | | | | |

Obr. 23: Výsledok optimalizácie

10.2 Generovanie alokácií

Ako už bolo uvedené v kapitole projektové plánovanie v systéme môžeme mať niekoľko typov alokácií.



Manuálne užívateľ môže pridať požiadavku na alokáciu nad rámec projektu alebo upraviť termín alokácie pracovnej pozície definovanej v projekte. Alokácie, u ktorých zamestnanec chce iba upraviť termín trvania musia mať vyplnený rovnaký názov projektu, pracovnú pozíciu a percentuálne priradenie aby systém vedel tieto dva záznamy spárovať.

Ďalším typom sú alokácie, ktoré sú vygenerované na základe požiadaviek definovaných na úrovni projektu.

Tieto požiadavky sa generujú v prípade spustenia analýzy a následne sa ich snaží systém za použitia optimalizačného procesu zaplniť.

Alokácie, ktoré sú takto generované nemusia ale vždy byť jedna k jednej s požiadavkami ktoré sú uvedené v projekte. Pri ich generovaní musíme vziať do úvahy alokácie, ktoré boli manuálne upravené užívateľom.

Na Obr. 24 môžeme vidieť požiadavku na 3 zamestnancov.

| | | | | | | | | |
|---|-------------------------------|---------------------------------|---------------------------------|-------------------|-----------------|--|------------------|--------------|
| Search | | | | | | | | |
| | Name ▾ Please enter | Region ▾ Please enter | Status ▾ Please enter | Start Date | End Date | Jobs | Employees | Notes |
|   | Project1 | EMEA | Active | 4/1/2016 | 6/30/2016 | 3 100% Configuration Engineer 100% Deal manager 100% x | | |







Obr. 24: Nastavenie projektu

Manuálne sme pridali 2 záznamy.

Prvý, so všetkými parametrami rovnakými ako v požiadavkách na projekte a upravili sme iba termín alokácie.

V druhom sme okrem termínu modifikovali i požiadavku na pracovnú dobu.

Ako môžeme vidieť na Obr. 25 systém situáciu správne vyhodnotil situáciu a vygeneroval iba 2 nové záznamy.

| | | | | | | | | | | | |
|--|---|---|--------------------------------------|--------------------------------|--------------------------------------|-----------------------------------|--|--|---|-------------------|-----------------|
| | | Project Name ▾ Please enter a value | Job ▾ Please enter a value | Skill ▾ Please enter | Knowledge % ▾ Please enter | Employee ▾ Please enter | Resource Location ▾ Please enter a value | Supply/Demand ▾ Please enter a value | Assignment % ▾ Please enter a value | Start Date | End Date |
|    | A | Project1 | Configuration Engineer | Deal manager | 100 % | Employee 4 | Offshore | Demand | 100 % | 4/1/2016 | 6/30/2016 |
|    | A | Project1 | Configuration Engineer | Deal manager | 100 % | Employee 2 | Offshore | Demand | 100 % | 4/1/2016 | 6/30/2016 |
|    | | Project1 | Configuration Engineer | Deal manager | 100 % | Employee 3 | Onshore | Supply | 90 % | 4/1/2016 | 4/30/2016 |
|    | | Project1 | Configuration Engineer | Deal manager | 100 % | Employee 6 | Offshore | Demand | 100 % | 4/1/2016 | 4/30/2016 |

Obr. 25: Výsledok optimalizácie

10.3 Časová náročnosť genetického algoritmu

Pre overenie časovej náročnosti výpočtu genetického algoritmu bolo prevedených niekoľko testov.

V súčasnej dobe má firma Vendavo s.r.o. okolo 200 zamestnancov a niekoľko desiatok projektov. Mnoho projektov je ale obsadených a zamestnancov alokovaných.

Preto pri testovaní časovej náročnosti nie je potrebné pracovať s príliš veľkým množstvom dát. Pár základných výsledkov testovania môžeme vidieť v Tabuľka 3:

| | Test 1 | Test 2 | Test3 |
|---------------------------|--------|--------|--------|
| Počet pracovných pozícií | 11 | 11 | 40 |
| Počet zamestnancov | 7 | 10 | 30 |
| Minimálny počet populácií | 4 | 1 | 33 |
| Maximálny počet populácií | 26 | 5 | 865 |
| Minimálny čas (ms) | 1828 | 1887 | 20154 |
| Maximálny čas (ms) | 4438 | 5165 | 250971 |

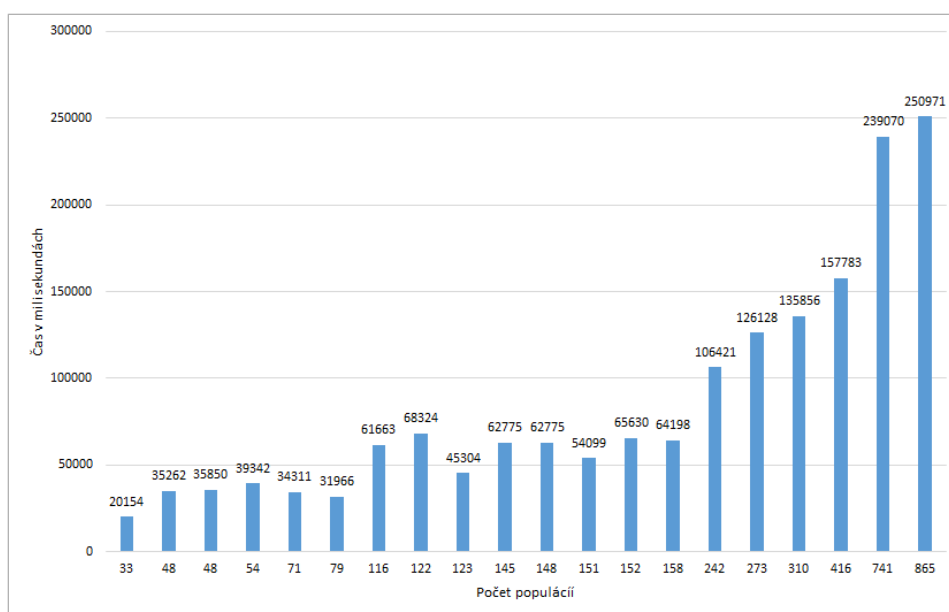
Tabuľka 3: Časová náročnosť nájdenia výsledku

Na Obr. 26 môžeme vidieť graf znázorňujúci akúsi nevyspytateľnosť genetických algoritmov. Graf znázorňuje spustenie genetického algoritmu, ktorý 20 krát za sebou hľadal optimálne riešenie obsadenia 40 pracovných pozícií pri dostupnosti 30 zamestnancov.

Na grafe môžeme vidieť, že v najhoršom prípade trvalo nájdenie optimálneho riešenia 4 minúty, kým v najlepšom prípade trvalo nájdenie optimálneho riešenia iba 20 sekúnd.

Aj keď je rozdiel medzi najlepším a najhorším časom pomerne veľký nájdenie riešenia v priebehu 4 minút je mnohonásobne lepším výsledkom, akého by sme dosiahli za použitia súčasného riešenia.

Ako bolo v práci už spomenuté v súčasnej dobe sa pre proces plánovania používa Excel a všetky priradovania na projekty sa robí manuálne bez podpory akéhokoľvek pomocného výpočtu či algoritmu.



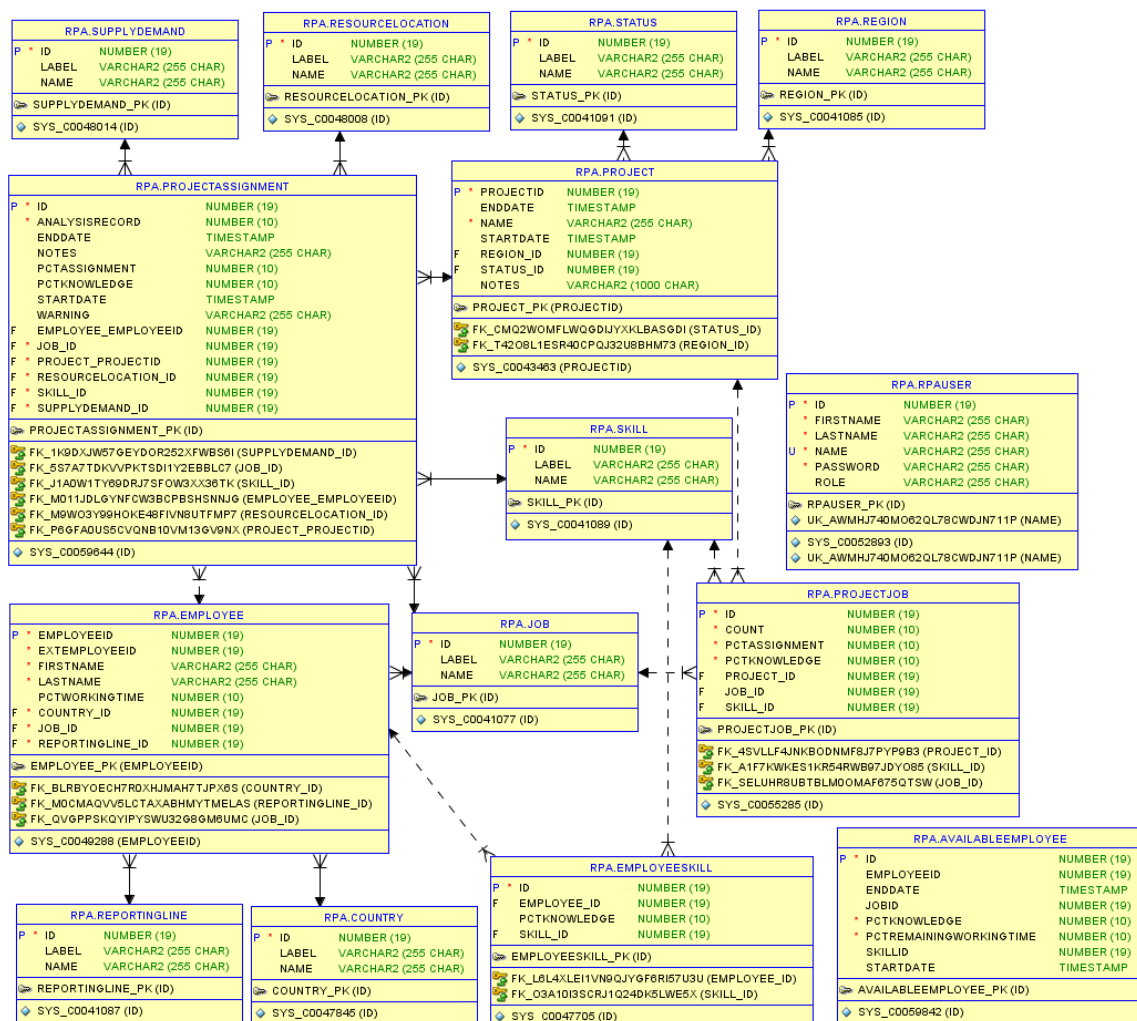
Obr. 26: Časová náročnosť obsadenia 40tich pracovných pozícií

11 Návrh a ďalšia implementácia

V nasledujúcej kapitole sa zameriame na dátový model aplikácie, ukážeme si základné use cases a priblížime si implementáciu genetického algoritmu.

11.1 ER diagram

Na Obr. 27 môžeme vidieť ER diagram, znázorňujúci celú štruktúru databázy. Na schéme je možné vidieť jednotlivé entity s výpisom atribútov.



Obr. 27: ER Diagram

11.2 Technológie

Kapitola je venovaná popisu použitých technológií. V priebehu implementácie som ich využila väčšie množstvo a tak sa zameriam len na najdôležitejšie z nich.

11.2.1 JavaScript

JavaScript je multiplatformný netypový interpretovaný skriptovací jazyk umožňujúci interaktívnu manipuláciu s DOM stromom. Spolu s HTML and CSS je považovaný za základný stavebný kameň tvorby obsahu v prostredí webu. Syntax je odvodená z jazyka C, zatiaľ čo sémantika je ovplyvnená programovacími jazykmi Self a Scheme. JavaScript sa používa aj v prostrediach mimo web, napríklad PDF dokumentoch a desktopových widgetoch. Nové a rýchlejšie JavaScriptové virtuálne stroje a platformy na nich postavené zvýšili popularitu JavaScriptu ako riešenia pre serverovú stranu webových aplikácií. Na klientskej strane je JavaScript tradične implementovaný ako interpretovaný jazyk, avšak novšie verzie webových prehliadačov umožňujú just-in-time kompiláciu. Je používaný pri vývoji počítačových hier, vytváraní desktopových a mobilných aplikácií, a programovaní na strane servera s použitím prostredí ako Node.js.

11.2.2 React JS

React je opensource knižnica vyvinutá Facebookom, určená na tvorbu user interface. Je to knižnica, ktorá v pomyselnom MVC predstavuje „V“ čiže view vrstvu. React na rozdiel od iných JavaScriptových knižníc preferuje HTML v kóde. Dáta putujú iba jedným smerom. React tým znižuje zložitosť kódu.

11.2.3 AJAX

AJAX (Asynchronous JavaScript and XML) je všeobecné označenie technológií vývoja interaktívnych webových aplikácií. AJAX tvorí HTML, CSS, DOM, JavaScript a XML/JSON.

Medzi výhody týchto technológií patrí fakt, že odstraňujú nutnosť opätovného načítania a prekreslenia stránok po každej operácii.

Vďaka AJAXu môžu webové aplikácie prijímať a odosielať dáta asynchrónne, bez toho, aby musela prebiehajúca komunikácia zasahovať do vzhľadu alebo chovania stránky.

Ilustráciu AJAX kódu, slúžiaceho pre načítanie všetkých zamestnancov môžeme vidieť na Výpis 1

```
var _searchEmployees = function() {  
  _state.loading = true;  
  EmployeeStore.emitChange();  
  $.ajax({
```

```

url: _props.employeesUrl+'?filterType='+_state.actualFilter.id+'&
    startDate='+dateFormatter(_state.startDate)+'&endDate='+
    dateFormatter(_state.endDate),
success: function(data) {
    _state.employees = data;
    _state.loading = false;
    EmployeeStore.emitChange();
},
error: function(xhr, status, err) {
    _state.loading = false;
    var messages = getErrorMessages(xhr);
    MessageActions.add_message_error(messages);
    EmployeeStore.emitChange();
}
});
};

```

Výpis 1: AJAX - načítanie všetkých zamestnancov

11.2.4 JSON

JSON (JavaScript Object Notation) je spôsob zápisu dát nezávislý na počítačovej platforme. Tento formát je pre človeka jednoduchý na čítanie i zápis. JSON vie pracovať s jednoduchými hodnotami, poľami hodnôt i s objektmi.

Názornú ukážku toho, ako môže vyzeráť dokument formátu JSON si ukážeme na zamestnancovi Employee 1, ktorého môžeme vidieť na obrázku Obr. 28 a následne JSON reprezentujúci tohto zamestnanca na Obr. 29

| | | | | | | | | |
|------|----------|---|---------------------------|------------------------|----------------|-----|---|---|
| 8014 | Employee | 1 | GSD CZ Program Management | Configuration Engineer | Czech Republic | 100 | % | Price manager 100% Deal manager 100% |
|------|----------|---|---------------------------|------------------------|----------------|-----|---|---|

Obr. 28: Zamestnanec Employee1

11.2.5 REST API

REST (Representational State Transfer) je architektúra rozhrania, navrhnutá pre distribuované prostredie.

Rozhranie REST je použiteľné pre jednoduchý prístup k dátam.

Táto architektúra definuje jednotný prístup pre získanie a manipuláciu s dátami v podobe 4 operácií CRUD.

```

▼ 0: {employeeId: 1, extEmployeeId: 8014,...}
  ► country: {id: 60, name: "CZ", label: "Czech Republic"}
    countryLabel: "Czech Republic"
    employeeId: 1
  ▼ employeeSkills: [{id: 7, skill: {id: 2, name: "PM", label: "Price manager"}, employeeId: 1, pctKnowledge: 100},...]
    ► 0: {id: 7, skill: {id: 2, name: "PM", label: "Price manager"}, employeeId: 1, pctKnowledge: 100}
    ► 1: {id: 1, skill: {id: 1, name: "DM", label: "Deal manager"}, employeeId: 1, pctKnowledge: 100}
    endDate: null
    extEmployeeId: 8014
    firstName: "1"
    fullName: "Employee 1"
  ► job: {id: 6, name: "CONFIG_ENGINEER", label: "Configuration Engineer"}
    jobLabel: "Configuration Engineer"
    lastName: "Employee"
    pctWorkingTime: 100
    projects: []
  ► reportingLine: {id: 4, name: "VCZ44910", label: "GSD CZ Program Management"}
    reportingLineLabel: "GSD CZ Program Management"
    startDate: null

```

Obr. 29: JSON zamestnanca Employee 1

Úrovne REST API

Nultá úroveň

Úroveň slúži pre prenos dát.

Funkcionalita nie je striktne viazaná iba na HTTP, ale v súčasnosti je to najčastejšie využívaný protokol.

Prvá úroveň – zdroje (Resources)

Namiesto toho, aby sa všetky požiadavky posielali na jeden centrálny bod, má každý zdroj práve jeden koncový bod, kde je možné ho dosiahnuť.

Napr. GET /employees vráti všetkých zamestnancov.

Druhá úroveň - HTTP Verbs

Medzi najznámejšie metódy patria:

- GET (Retrieve) – získanie dát

Každý zdroj má svoj vlastný identifikátor (URI), podľa ktorého metódou GET získavame dáta tohto zdroja.

GET /api/employees/ nám vráti všetkých zamestnancov

GET/api/employees/1 vráti zamestnanca s ID 1

- POST (Create) – vytvorenie dát

Vo chvíli volania metódy ešte nepoznáme URI pridávaného zdroja.

Preto sa táto metóda volá bez akéhokoľvek identifikátoru.

POST /api/employees/ vytvorí nový zdroj. Pri vytváraní v tele metódy potom definujeme hodnoty zdroja

- PUT (Update) - modifikácia dát

Na rozdiel od metódy POST u tejto metódy voláme konkrétne URI zdroja, ktorý chceme modifikovať. V tele metódy potom predáme nové hodnoty. PUT /api/employees/1

- DELETE (Delete) – zmazanie dát

DELETE /api/employees/1 zmaže zamestnanca s ID 1.

V praxi môžu s touto metódou nastať problémy, pretože HTML formuláre a niektoré HTTP nástroje sú obmedzené iba na používanie metód POST a GET.

Tento problém sa rieši napríklad tak, že sa odošle metóda POST s parametrom, ktorý nesie informáciu o tom, že má byť použitá metóda DELETE.

Tretia úroveň – Hypermedia Controls

Posledná úroveň je známa ako HATEOAS (Hypertext as the Engine of Application State). Zjednodušene ide o to, že poznáme iba základný koncový bod, ktorý okrem dát vracia odkazy na ďalšie zdroje, ktoré sa rekurzívne odkazujú na ďalšie zdroje.

Pre výmenu dát využíva REST niekoľko jednoduchých formátov. Medzi tieto formáty patrí napríklad XML, RSS, ATOM a JSON.

11.3 Integrácia s ERP systémom

Jednou z požiadaviek na nový systém bola implementácia importu dát z ERP systému. Primárnou funkciou importu je vytváranie, prípadne modifikácia zamestnancov.

Okrem zamestnancov budeme v rámci importu pracovať s dvoma číselníkmi, ktorými sú oddelenie a pracovná pozícia.

Vstupom pre import dát bude súbor dát vo formáte CSV.

Tento súbor musí obsahovať nasledujúce informácie o zamestnancovi:

- ERP ID zamestnanca,
- meno a priezvisko zamestnanca,
- kód pracovnej pozície,
- názov pozície,
- kód oddelenia,
- názov oddelenia,
- krajina, v ktorej je zamestnanec zamestnaný,
- pracovný úväzok zamestnanca vyjadrený v percentách.

| Column 1 | Column 2 | Column 3 | Column 4 | Column 5 | Column 6 | Column 7 | Col... |
|----------|------------|-------------------|-------------------|------------|---------------------------|----------|--------|
| Employee | | Job Id | Job | Reporti... | Reporting Line Unit | Country | Ass... |
| 8000 | Employee 1 | QA_ENGINEER | Quality Assura... | VCZ44310 | GSD CZ Quality Assurance | Germany | 100 |
| 8001 | Employee 2 | QA_ENGINEER | Quality Assura... | VCZ44310 | GSD CZ Quality Assurance | Germany | 100 |
| 8002 | Employee 3 | ENGINEER | Engineer | VCZ31100 | Engineering Development | Germany | 100 |
| 8007 | Employee 4 | LEAD_SOL_ENGINEER | Lead Solution ... | VCZ44910 | GSD CZ Program Management | Germany | 100 |
| 8010 | Employee 5 | SUPPORT_ENGINEER | Support Engineer | VCZ36000 | Global Support | Germany | 100 |

Obr. 30: CSV

Vstupný súbor má pevne danú štruktúru, znázornenú na Obr. 30

Celkovú funkcionálnu spojenú s importom popisuje use case uvedený v Tabuľka 5:

| | |
|--------------------------|--|
| Scope | System |
| Level | User Goal |
| Primary Actor | Staffing manažér |
| Stakeholders & Interests | Staffing manažér – Import dát z ERP systému |
| Precondition | CSV súbor s vyexportovanými zamestnaneckými dátami z ERP systému |
| Minimal Guarantees | Výpis chybovej správy o neúspešnom priebehu importu |
| Success Guarantees | Import nových zamestnancov, oddelení a pracovných pozícií, prípadne modifikácia stávajúcich |
| Trigger | Spustenie import adaptéru |
| Main Success Scenario | <ol style="list-style-type: none"> 1. Užívateľ spustí import. 2. Systém importuje záznam po zázname a rozloží ho na relevantné elementy. 3. Systém do databáze uloží novú alebo modifikovanú pracovnú pozíciu. 4. Systém do databáze uloží nové alebo modifikované pracovné oddelenie. 5. Systém do databáze uloží nového alebo modifikovaného zamestnanca. 6. Po ukončení importu systém zobrazí importované záznamy v aplikácii. |

| | |
|------------|---|
| Extensions | <ol style="list-style-type: none"> 3. (a) Daná pracovná pozícia neexistuje. Systém ju vytvorí a vloží do databáze. (b) Daná pracovná pozícia existuje. Systém modifikuje stávajúcu pozíciu. 4. (a) Dané oddelenie neexistuje. Systém ho vytvorí a vloží do databáze. (b) Dané oddelenie existuje. Systém modifikuje stávajúce oddelenie. 5. (a) Daný zamestnanec neexistuje. Systém ho vytvorí a vloží do databáze. (b) Daný zamestnanec existuje. Systém modifikuje stávajúceho zamestnanca. |
|------------|---|

Tabuľka 4: Import dát z ERP systému

11.4 Kalendárový komponent

Tabuľkové zobrazenie nám poskytuje podrobný prehľad o projektoch, zamestnancoch a ich alokácii na projekt.

Čo nám ale toto zobrazenie neumožňuje je rýchly a jednoduchý prehľad o začiatku a konci alokácie, prípadne o dĺžke projektu.

Tento nedostatok bol motiváciou k implementácii alebo použitiu už existujúceho grafického rozhrania ktoré by nám tieto možnosti poskytlo.

Pre účely vytvorenia požadovaného prehľadu bol zvolený komponent React calendar timeline. Dokumentáciu komponentu nájdeme na webe <https://www.npmjs.com/package/react-calendar-timeline>.

Komponent umožňuje vytvoriť harmonogram.

Položky vodorovnej osy x rozdeľujú harmonogram na jednotlivé časové úseky. Komponent nám umožňuje definovať časové úseky dĺžky od minúty až po mesiac.

V našom prípade používame možnosť rozdelenia na mesiace, je možné ale použiť i rozdelenie podľa hodín.

Potenciál zobrazenia s vyššou granularitou vidím napríklad u plánovania zmien zamestnancov v spoločnostiach s trojzmennou prevádzkou.

Na zvislej ose y potom definujeme položky, pre ktoré chceme harmonogram zobrazíť. V našom prípade ide o projekty a zamestnancov.

```
var projectAssignments = this.state.projectAssignments;
if (!isNull(projectAssignments)){
  projectAssignments.forEach(function(projectAssignment) {
    var start = dateFormatter(projectAssignment.startDate,null);
    var end = dateFormatter(projectAssignment.endDate,null);
    var itemClassName = "standardItem";
    if (start!= "" && end!= ""){
      itemLabel = "";
      groupLabel=projectAssignment.projectName;
      groupLabel = groupLabel.concat(" ",projectAssignment.pctAssignment,"%")
    );
      var group = {"id":i, "title":groupLabel};

      groups.push(group);

      var endDate = new Date(projectAssignment.endDate);
      endDate.setDate(endDate.getDate() + 1);
      if (projectAssignment.analysisRecord != 0) {
        itemClassName = "analysisItem";
      }

      if (!isNull(projectAssignment.employee)) {
        var itemLabel = projectAssignment.employee.fullName;
      } else {
        var itemLabel = projectAssignment.job.label;
        itemClassName = "emptyItem";
      }

      var item = {"id":j,"title":itemLabel, "group":i,start_time:
        projectAssignment.startDate,end_time:endDate, "className":
        itemClassName};
      items.push(item);
      j=j+1;
      i=i+1;
    }
  });
}
```

```

<Timeline
  groups={groups}
  items={items}
  canMove={false}
  defaultTimeStart={moment()}
  defaultTimeEnd={moment().add(1, 'month')}
/>

```

Výpis 2: TimeLine

Výsledkom vyššie zmienenej implementácie môže byť napríklad harmonogram, ktorý vidíme na Obr. 31, kde:

- modré položky označujú schválené alokácie,
- zelené položky sú výsledkom What-if analýzy, znázorňujúce prípad, kedy sa nám podarilo úspešne alokovať človeka na projekt,
- červené položky upozorňujúce na fakt, že sa nám nepodarilo nájsť zamestnanca, ktorého by sme mohli na projekt priradiť.

| | 2016 | | | | | | | | | |
|---------------|------|------------|------------------------|-----|------------------------|------------|-----|-----|-----|--|
| | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | |
| Project1 100% | | Employee 1 | | | | | | | | |
| Project1 100% | | Employee 2 | | | | | | | | |
| Project2 100% | | | Configuration Engineer | | | | | | | |
| Project2 50% | | | Configuration Engineer | | | | | | | |
| Project3 50% | | | | | Employee 2 | | | | | |
| Project4 100% | | | | | Configuration Engineer | | | | | |
| Project5 100% | | | | | | Employee 1 | | | | |

Obr. 31: Harmonogram

11.5 Priradovanie zamestnancov na projekty

Akýmsi základom aplikácie slúžiacej k plánovaniu ľudských zdrojov je nástroj, ktorý slúži k priradovaniu ľudí na projekty. Na základe jednotlivých priradení je ďalej možné zobrazovať napríklad grafický prehľad alokácie zamestnancov, prehľad zamestnancov bez alokácie a podobne.

K vytváraniu nových alokácií nám slúži jednoduchý tabuľkový prehľad stávajúcich alokácií a ďalej jednoduchý modálny dialóg pre vytvorenie novej alokácie. Ukážku dialógu môžeme vidieť na Obr. 32

Project Assignment

Project* Project1 (05/01/2016 - 07/31/2016)

Job* Quality Assurance Engineer

Skill* Deal manager 100 %

Employee Employee 4

Resource Location* Onshore

Supply/Demand* Supply

Assignment %* 100 %

Duration 4/1/2016 4/30/2016

Notes

Save **Cancel**

Obr. 32: Vytvorenie alokácie

Proces priradenia zamestnanca na projekt je znázornený v Tabulka 5:

| | |
|--------------------------|--|
| Scope | System |
| Level | User Goal |
| Primary Actor | Staffing manažér |
| Stakeholders & Interests | Staffing manažér – Vytvorenie novej alokácie |
| Precondition | Existuje projekt, na ktorý môžeme priradiť zamestnanca |
| Minimal Guarantees | Výpis chybovej správy |
| Success Guarantees | Priradenie zamestnanca na projekt |
| Trigger | Užívateľ spustí proces pridávania užívateľa na projekt |

| | |
|-----------------------|---|
| Main Success Scenario | <ol style="list-style-type: none"> Užívateľ vyplní nasledujúce údaje. <ul style="list-style-type: none"> názov projektu, názov pracovnej pozície, názov znalosti a jej percentuálnu úroveň, meno zamestnanca, názov umiestnenia zamestnanca, supply/demand, percentuálne priradenie na projekt, trvanie alokácie, poznámky. Systém overí zadané údaje. Systém vygeneruje unikátny identifikátor pre vytváraný záznam a uloží ho do databáze. |
| Extensions | <ol style="list-style-type: none"> <ol style="list-style-type: none"> Zadané dáta nie sú validné – systém vyzve užívateľa k opätovnému zadaniu potrebných údajov – návrat ku kroku 1. Zadané dáta sú v poriadku – pokračujeme krokom 3. |

Tabuľka 5: Priradenie zamestnanca na projekt

12 Záver

Cieľom práce bol návrh a implementácia aplikácie pre plánovanie ľudských zdrojov.

Výsledným riešením sa mi podarilo splniť všetky ciele, ktoré boli po aplikácii požadované. Aplikácia bola rozdelená do štyroch základných modulov:

- Správa zamestnancov

V správe zamestnancov máme k dispozícii všetky dôležité informácie o zamestnancovi. Vďaka naimplementovanému filtrovaniu je možné veľmi rýchlo získať informácie o zamestnancoch, ktorí sú v požadovanom období priradení na projekte alebo naopak nie sú nikde alokovaní. Modul zároveň podporuje import dát z ERP systému a to vo formáte CSV. Poslednou časťou tohto modulu je kalendárový komponent zobrazujúci harmonogram alokácii zamestnancov.

- Správa projektov

Modul nám poskytuje prehľad všetkých projektov firmy. Je možné ich opäť filtrovať podľa rôznych kritérií. Vďaka tomu môžeme získať rýchly prehľad napríklad o aktuálnych i budúcich projektoch, či projektoch bez zamestnancov.

- Správa alokácií

Správa alokácii je najdôležitejším modulom celej aplikácie. Okrem možnosti manuálnej správy dát umožňuje What-if analýzu. Vďaka analýze môžeme simulovať rôzne potenciálne scenáre a predísť tým prípadným problémom s nedostatkom zamestnancov. What-if analýza však neslúži iba k simulácii. Je možné využiť ju v procese plánovania, pretože alokácie, ktoré vzniknú je možné jednoducho schváliť ako platné alokácie. Pre optimalizáciu rozmiestnenia zamestnancov na projekty bol použitý genetický algoritmus, ktorý behom krátkej chvíle dokáže nájsť optimálne rozloženie.

- Administrácia

V poslednom module aplikácie máme prístup k jednotlivým číselníkom využívaným v rámci celej aplikácie. Zároveň obsahuje správu užívateľov, ktorí môžu s aplikáciou pracovať.

Výsledná aplikácia výrazne zjednodušuje a zrýchľuje proces plánovania. Dokáže nájsť riešenia, ktoré nemusia byť zrejmé a tým šetrí čas a peniaze.

V budúcnosti sa počíta s ďalším vývojom aplikácie. Proces plánovania je možné neustále vylepšovať. Do budúcnosti by bolo vhodné pridať možnosť spravovať historické dáta. Na ich základe by potom mohol genetický algoritmus priradovať ľudí na projekty na základe prechádzajúcich skúseností. Potenciál tohto riešenia vidím u projektov minulých či aktuálnych, ktoré sa rozhodnú podpísať kontrakt na ďalšiu fázu vývoja. Na projekty potom môžeme pridať ľudí, ktorí majú s daným projektom skúsenosti, čo môže proces ďalšieho vývoja značne urýchliť.

Táto práca bola pre mňa obrovským prínosom. Doteraz som mala skúsenosti iba s vývojom backendovej časti aplikácie a vďaka tejto práci som si vyskúšala i prácu s frontendom.

Literatura

- [1] Hordějčuk Ing. Vojtěch. Genetický algoritmus [online]. Dostupné z <http://voho.cz/wiki/geneticky-algoritmus>
- [2] Wikipedie [online]. Dostupné z https://cs.wikipedia.org/wiki/Genetický_algoritmus
- [3] Vendavo CZ [online]. Dostupné z <http://vendavo.cz/>
- [4] Genetické algoritmy [online]. Dostupné z <https://akela.mendelu.cz/xpopelka/cs/ui/ucici>
- [5] Komínek Bc. Jan. Heuristické algoritmy pro optimalizaci. Brno, 2012. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství

A Príloha na DVD

| Cesta | Názov súboru | Popis |
|------------------|-------------------|--|
| /doc/... | zadani.pdf | Zadanie diplomovej práce |
| /doc/... | gav101_DP.pdf | Diplomová práca |
| /src/... | - | Zdrojové kódy aplikácie |
| /attachments/... | create_schema.sql | DDL pre vytvorenie DB schémy |
| /attachments/... | init_db.sql | DDL pre vytvorenie databázovej štruktúry |
| /attachments/... | countries.sql | DDL pre nahranie krajín |
| /attachments/... | import_sample.csv | Import - ukážkový súbor |

Tabuľka 6: Obsah CD